Chương 1. ĐẠI SỐ BOOL VÀ CỔNG LOGIC

❖ ***Đề Cương:***

- Biến logic, hằng logic, hàm logic và mạch logic.
- Các hàm logic cơ bản và các cổng logic.
- Mô tả mạch logic bằng hàm logic.
- Xây dựng mạch logic từ biểu thức logic hay hàm logic.
- Xác định ngõ xuất của mạch logic ứng với một bộ giá trị ngõ nhập.
- Xác định giá trị của hàm logic ứng với một bộ giá trị biến logic.
- Các phép toán logic và cổng logic được xây dựng thêm.
- Các định lý của đại số logic.
- Các vi mạch chứa các cổng logic.
- Các phương pháp biểu diễn hàm logic theo các dạng chuẩn
- Rút gọn bằng phương pháp đại số
- Rút gọn bằng phương pháp biểu đồ Karnaugh- dạng chuẩn tuyển
- Rút gọn bằng phương pháp biểu đồ Karnaugh- dạng chuẩn hội
- Rút

❖ ***Mục Đích:***

Sau khi hoàn thành chương này, bạn phải nắm được kiến thức:

- Hàm logic, cổng logic và mạch logic.
- Các định lý của đại số logic.
- Các vi mạch chứa cổng logic.
- Xác định ngõ xuất của mạch logic ứng với một bộ giá trị ngõ nhập.
- Các phương pháp biểu diễn hàm logic.
- Rút gọn hàm logic bằng phương pháp đại số và bằng biểu đồ Karnaugh.

❖ ***Các Thuật Ngữ Tiếng Anh:***

- Truth table: bảng thực trị
- Logic variable: biến logic
- Logic constant: hằng logic
- Logic expression: biểu thức logic
- Gate: cổng logic
- Input: ngõ nhập
- Output: ngõ xuất
- Active Low: tích cực mức thấp
- Active High: tích cực mức cao
- Minterm: dạng chuẩn tuyển (tổng các tích)
- Maxterm: dạng chuẩn hội (tích các tổng)

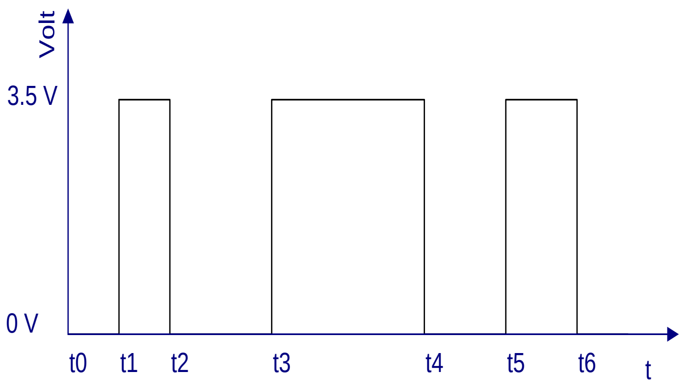
1.1- Biến Logic, Hằng Logic, Hàm Logic Và Mạch Logic

Cuối thế kỷ 19, nhà toán học Bool đã sáng lập ra một ngành toán học mang tên ông là ***Đại Số Bool*** hay cũng được gọi là ***Đại Số Logic***. Đại số Bool nghiên cứu về hệ thống nhị phân. Đây là cơ sở và cũng là công cụ toán học cho mọi lĩnh vực có liên quan đến hệ thống kỹ thuật số.

- Cho 1 tập hợp X = {0, 1}, A được gọi là biến logic nếu A∈X, có nghĩa là A chỉ nhận 1 trong 2 giá trị: A = 0 hoặc A = 1.
- Trong logic mệnh đề, đây là hai giá trị ***đúng*** hay ***sai***. Trong kỹ thuật thì nó tương ứng với mức điện thế 0 hay 1.
- Trong logic mệnh đề, nếu lấy giá trị 0 là sai và giá trị 1 là đúng, thì ta gọi là ***logic dương***. Nếu chọn giá trị 1 là sai và giá trị 0 là đúng thì ta gọi là ***logic âm***.
- Logic dương trong kỹ thuật (họ TTL), nếu chọn mức điện thế 0V-0.8V là giá trị 0, và 2V-5V là giá trị 1. Logic âm trong kỹ thuật, nếu chọn mức điện thế 0V-0.8V là giá trị 1, và 2V-5V là giá trị 0.
- Trong kỹ thuật số hiện nay, hầu hết mạch được thiết kế và chế tạo theo ***logic dương***, nên người ta cũng hay dùng từ ***mức cao*** chỉ giá trị 1 và từ ***mức thấp*** chỉ giá trị 0.
- Và ***giá trị 0*** và ***1*** này là ***ký hiệu logic***, không phải là ***con số 0*** và ***số 1*** trong ***hệ thống số nhị phân*** đã đề cập ở những chương trước. Từ giờ trở đi, ta sẽ dùng các chữ cái A, B, C... là ký hiệu đại số cho biến logic.

1.1.1- Biến logic

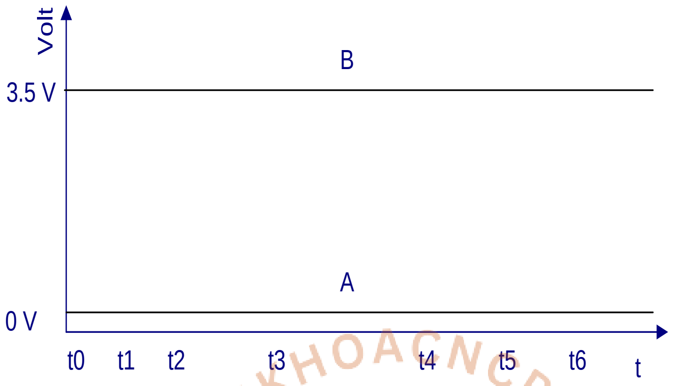
A được gọi là ***biến logic***, có nghĩa là nó có thể thay đổi giá trị là 0 hay 1 theo thời gian



Hình 1. Giản đồ thời gian của biến A

1.1.2- Hằng logic

Biến logic A được gọi là ***hằng logic*** khi nó bằng 0 (được gọi là ***hằng 0***) hay bằng 1 (được gọi là ***hằng 0***) trong toàn bộ thời gian. Như vậy, hằng logic là trường hợp đặc biệt của biến logic.



A là hằng 0, B là hằng 1

Hình 2. Giản đồ thời gian của hằng A và hằng B.

1.1.3- Hàm Logic Và Mạch Logic

- Một hàm x=f(A_1, A_2, ...,A_n) được gọi là ***hàm logic n biến***, nếu A_1, A_2, ...,A_n là biến logic và x ∈ X với X= {0, 1}.

- Vế phải của hàm là một ***biểu thức logic***. Vế trái là một biến logic.

- Biểu thức logic (biểu thức Bool) là một biểu diễn đại số với các toán hạng là biến logic và các toán tử là ***phép toán logic*** căn bản.

- Từ một biểu thức logic, ta xây dựng được một ***mạch logic*** tương ứng và ngược lại, từ một mạch logic tương ứng ta có thể biểu diễn bằng một biểu thức logic. Các ngõ nhập của mạch logic tương ứng với các biến logic, ngõ xuất tương ứng với giá trị của hàm số.

- Một mạch logic tương ứng với một phép toán logic căn bản của đại số Bool thì được gọi là ***cổng logic***.

- Để thể hiện hoạt động của một hàm logic hay một mạch logic, người ta thường liệt kê giá trị của hàm (ngõ xuất của mạch logic) tương ứng với một bộ giá trị của biến (hay ngõ nhập của mạch logic). Bảng liệt kê toàn bộ giá trị của hàm theo biến (hay ngõ xuất theo ngõ nhập của mạch logic) được gọi là bảng thực trị.

1.1.4- Bảng thực trị

- Bảng thực trị cho một hàm $x=f(A_1, A_2, ..., A_n)$ là một bảng gồm N+1 cột và 2^N dòng. Thể hiện toàn bộ giá trị của hàm tương ứng với các tổ hợp biến.

  Ví dụ: cho 2 hàm x = f(A, B); y = g(A, B, C) ta lập bảng thực trị như sau:

A	B	x
0	0	1
0	1	0
1	0	1
1	1	0

C	B	A	y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Bảng 1. Bảng thực trị của hàm x = f(A, B) và hàm y = g(A, B, C)

- Nhận xét: bảng thực trị cho hàm hai biến x = f(A, B) có $2^2 = 4$ dòng và 2 + 1 = 3 cột. và hàm ba biến y = g(A, B, C) có $2^3 = 8$ dòng và 3 + 1 = 4 cột.

- Bảng thực trị thể hiện mối quan hệ giữa hàm và biến bằng cách lập bảng liệt kê tất cả bộ giá trị của biến số và hàm số. Ưu điểm là trực quan, nếu cho trước một bộ giá trị của biến logic, ta biết ngay giá trị hàm số tương ứng.

1.2- Các Phép Toán Logic Cơ Bản Và Các Cổng Logic

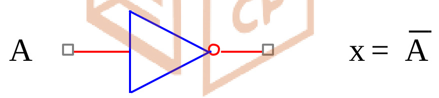
1.2.1- Phép toán NOT và cổng NOT

- Phép toán NOT là hàm một biến x = f(A).
- Ký hiệu đại số: $x = \overline{A}$ (đọc là x bằng NOT A, hay x bằng A đảo, hay x bằng A bù).
- Phép toán NOT là một phép toán cơ bản của đại số Bool, nó hoạt động theo bảng thực trị sau:

A	$x = \overline{A}$
0	1
1	0

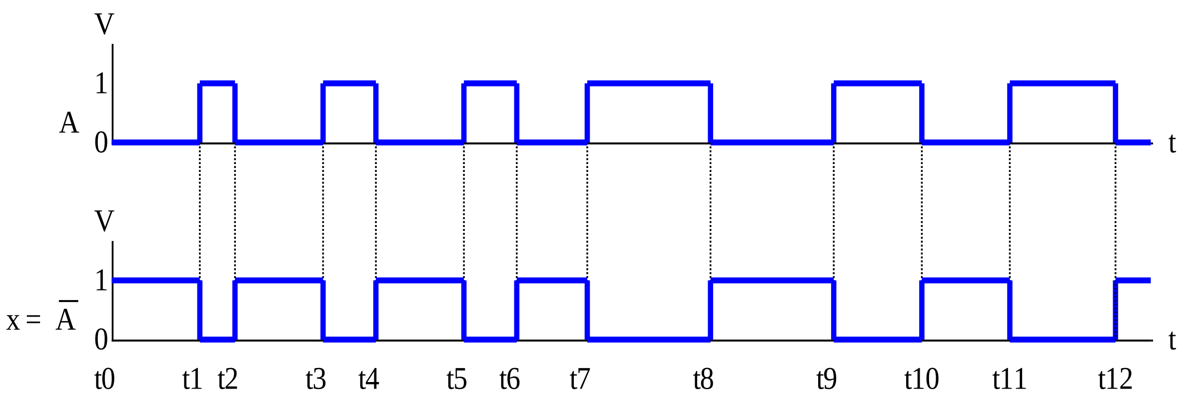
Bảng 2. Bảng thực trị của phép toán NOT

- Cổng NOT (hay còn gọi là ***cổng đảo***) là mạch số thực hiện phép toán NOT, có 1 ngõ nhập và 1 ngõ xuất và có ký hiệu cổng như sau:



Hình 3. Ký hiệu cổng NOT

- Giản đồ thời gian của biến A và hàm $x = \overline{A}$ được biểu diễn như hình sau:



Hình 4. Giản đồ thời gian của biến A và hàm $x = \overline{A}$

- Để vẽ giản đồ của $x = \overline{A}$, ta nhận thấy nó, luôn luôn ngược lại với A.

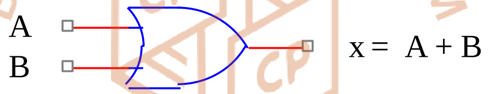
1.2.2- Phép toán OR và cổng OR

- Phép toán OR là hàm hai biến x = f(A, B).
- Ký hiệu đại số: x = A+B (đọc là x bằng A or B).
- Đây là phép toán cơ bản của đại số Bool, nó hoạt động theo bảng thực trị sau:

B	A	x = A + B
0	0	0
0	1	1
1	0	1
1	1	1

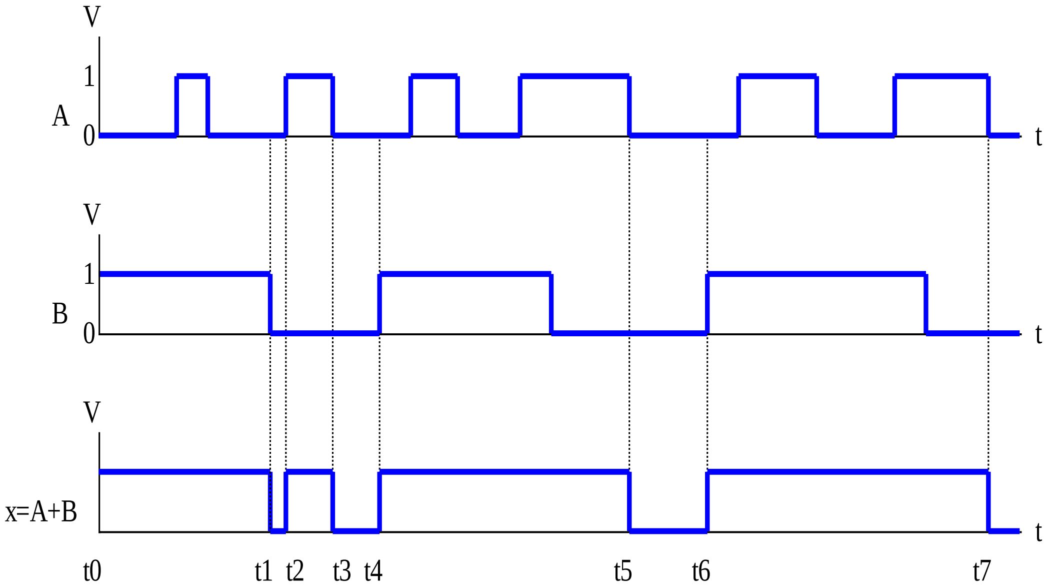
Bảng 3. Bảng thực trị của phép toán OR

- Cổng OR là mạch số thực hiện phép toán OR, có 2 ngõ nhập và 1 ngõ xuất và có ký hiệu cổng như sau:



Hình 5. Ký hiệu cổng OR

- Lưu ý: mặc dù là dùng ký hiệu (+), nhưng đây không phải là phép cộng nhị phân thông thường, mà là phép OR logic, nên có một số tác giả đã sử dụng ký hiệu khác thay cho ký hiệu (+) là ký hiệu (V). Nhưng hầu hết các tài liệu dùng trong kỹ thuật số đều dùng ký hiệu +. Nên trong tài liệu này vẫn dùng ký hiệu + biểu diễn cho phép OR.
- Giản đồ thời gian của hai biến A, B và hàm x=A+B được biểu diễn như hình sau:



Hình 6. Giản đồ thời gian của biến A, B và hàm x=A+B

- Để vẽ được giản đồ thời gian của x = A + B, ta chỉ cần quan tâm đến những thời điểm A và B đồng thời bằng 0 ⇒ x =0, trong trường hợp này là các thời điểm t1→t2, t3→t4, t5→t6, t7→..., các trường hợp khác x đều bằng 1.

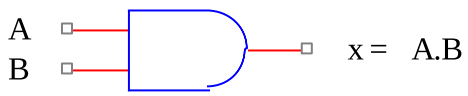
1.2.3- Phép toán AND và cổng AND

- Phép toán AND là hàm hai biến x = f(A, B).
- Ký hiệu đại số: x = A.B (đọc là x bằng A and B).
- Đây là phép toán cơ bản của đại số Bool, nó hoạt động theo bảng thực trị sau:

B	A	x = A.B
0	0	0
0	1	0
1	0	0
1	1	1

Bảng 4. Bảng thực trị của phép toán AND

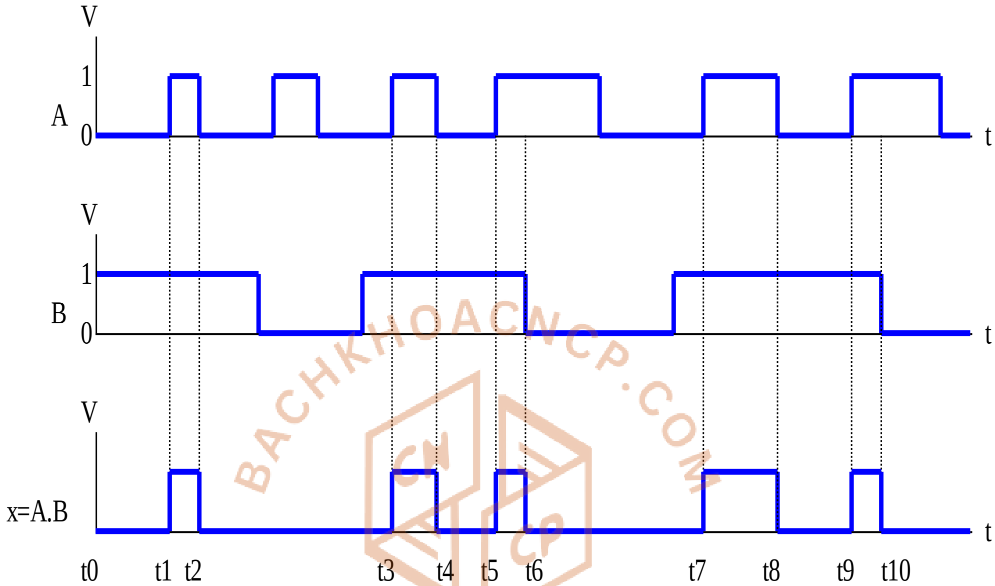
- Cổng AND là mạch số thực hiện phép toán AND, có 2 ngõ nhập và 1 ngõ xuất và có ký hiệu cổng như sau:



Hình 7. Ký hiệu cổng AND

- Lưu ý: mặc dù là dùng ký hiệu (.), nhưng đây không phải là phép nhân nhị phân thông thường, mà là phép AND logic. Nhưng lại có sự trùng hợp là phép AND logic hoàn toàn giống như phép nhân thông thường của 2 số nhị phân.

- Giản đồ thời gian của hai biến A, B và hàm x=A.B được biểu diễn như hình sau:



Hình 8. Giản đồ thời gian của biến A, B và hàm x=A.B

- Để vẽ được giản đồ thời gian của x = A . B, ta chỉ cần quan tâm đến những thời điểm A và B đồng thời bằng 1 ⇒ x = 1, trong trường hợp này là các thời điểm t1→t2, t3→t4, t5→t6, t7→t8, t9→t10, các trường hợp khác x đều bằng 0.

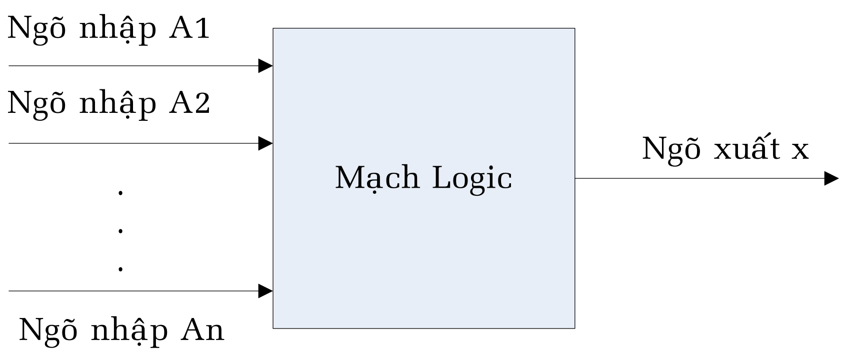
1.3- Mô Tả Mạch Logic Bằng Hàm Logic

- Thứ tự ưu tiên của các phép toán:

  Phép toán NOT có độ ưu tiên cao nhất, kế đến là phép toán AND, cuối cùng là phép toán OR.

  Nếu hai phép toán cùng độ ưu tiên , thì thực hiện theo thứ tự từ trái sang phải.

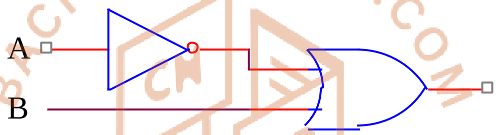
- Cho sơ đồ khối một mạch logic sau gồm một ngõ xuất x và n ngõ nhập đánh số từ A_1, A_2, ..., A_n. Mạch này sẽ được biểu diễn bằng hàm logic n biến x = f(A_1, A_2, ..., A_n).



Hình 9. Sơ đồ khối một mạch logic

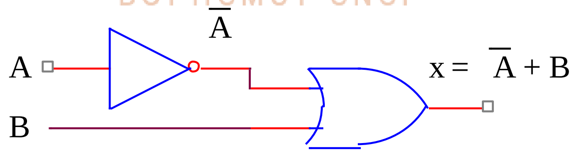
- ➢ Ta có thể xây dựng một hàm logic bất kỳ chỉ với 3 phép toán NOT, OR, AND đã đề cập. Hay nói cách khác ta có thể thiết kế một mạch số bất kỳ chỉ sử dụng 3 loại cổng NOT, OR, AND.

- ➢ Cho một số mạch logic cụ thể và mô tả chúng bằng hàm logic như sau:



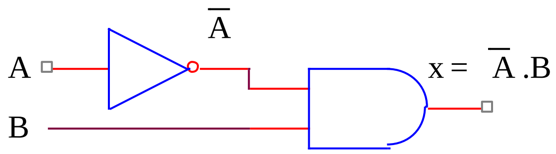
Hình 10. Sơ đồ một mạch logic dùng cổng NOT và cổng OR

Ta lần lượt ghi các hàm logic sau mỗi cổng như hình dưới, và ngõ xuất sau cùng chính là hàm logic của mạch logic đã cho.

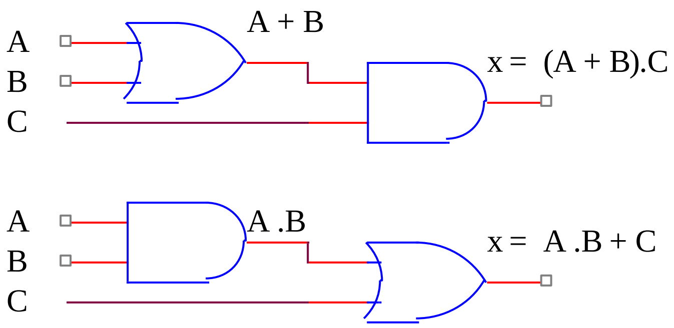


Hình 11. Hàm logic cho mạch dùng cổng NOT và cổng OR

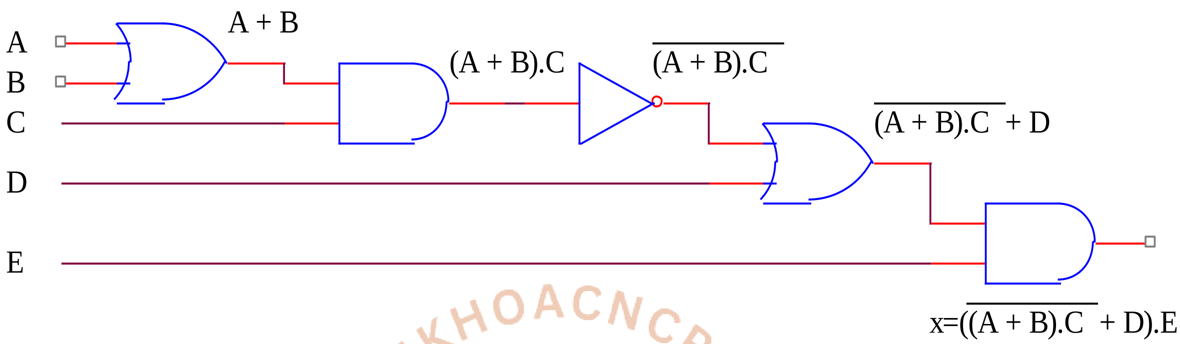
Tương tự cho những mạch logic đã cho ở dưới:



Hình 12. Hàm logic cho mạch dùng cổng NOT và cổng AND



Hình 13. Hàm Logic cho mạch dùng cổng OR và cổng AND

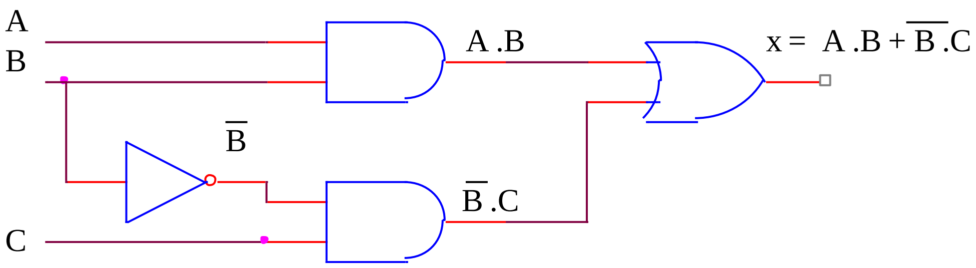


Hình 14. Mạch dùng cả 3 loại cổng OR. AND và NOT

1.4- Xây Dựng Mạch Từ Biểu Thức Logic Hay Hàm Logic

Từ một biểu thức logic hay hàm logic cho trước, ta xây dựng mạch logic bằng cách thay thế các phép toán logic bằng cổng logic tương ứng. như trong các ví dụ sau:

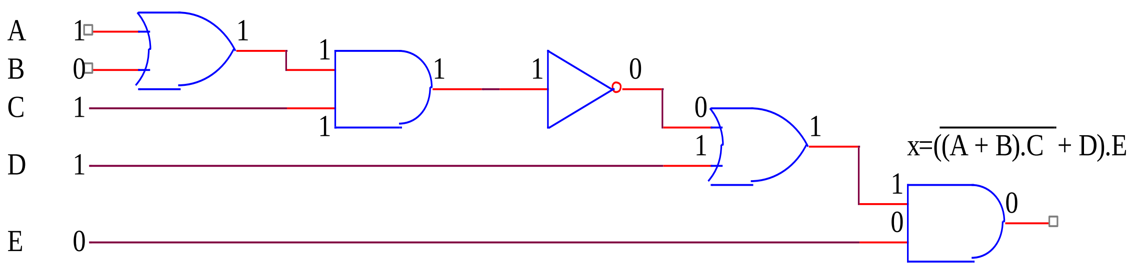
- Ví dụ: $x = A.B + \overline{B}.C$
- Ta xây dựng sơ đồ mạch tương ứng:



Hình 15. Sơ đồ mạch tương ứng từ hàm logic đã cho

1.5- Xác Định Ngõ Xuất Của Mạch Logic Ứng Với Một Bộ Giá Trị Ngõ Nhập

Cho một mạch logic như hình dưới, ứng với một bộ giá trị của ngõ nhập là (A,B,C,D,E) = (1,0,1,1,0) ta phải xác định giá trị ngõ xuất bằng cách lần lượt thay thế giá trị cụ thể của từng ngõ nhập. Lần lượt, tính ngõ xuất cho mỗi cổng, cho đến giá trị cuối cùng là x = 0 tương ứng với bộ giá trị đã cho.



Hình 16. Cách tính ngõ xuất của mạch logic ứng với một bộ giá trị ngõ nhập

1.6- Xác Định Giá Trị Của Hàm Logic Ứng Với Một Bộ Giá Trị Biến Logic

Cho một hàm logic $x = \overline{\overline{A}.B} + \overline{\overline{B}.C} + \overline{C+D}$ với bộ giá trị của biến logic là (A,B,C,D) = (1,0,0,1) ta tính giá trị tương ứng của x bằng cách thay tên biến A,B,C,D bằng các giá trị logic cụ thể là 1,0,0,1:

$x = \overline{\overline{1}.0} + \overline{\overline{0}.0} + \overline{0+1} = \overline{0.0} + \overline{1.0} + \overline{0+1} = \overline{0} + \overline{0} + \overline{1} = 1 + 1 + 0 = 1.$

1.7- Các Phép Toán Logic Và Các Cổng Logic Được Xây Dựng Thêm

Trong nhu cầu thực tiễn, người ta xây dựng thêm 4 phép toán khác và chúng cũng được đưa vào nhóm phép toán cơ bản mở rộng. Đó là các phép toán NOR, NAND, EX-OR, EX-NOR. Với 4 phép toán mới thêm vào là 4 loại cổng mang tên tương ứng.

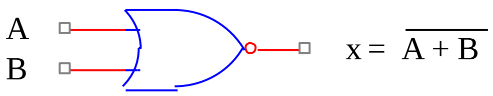
1.7.1- Phép toán NOR và cổng NOR

- Phép toán NOR là hàm hai biến x = f(A, B).
- Ký hiệu đại số: $x = \overline{A+B}$ (đọc là x bằng A nor B).
- Đây là phép toán dựa trên hai phép toán là NOT và OR, nó hoạt động theo bảng thực trị sau:

B	A	$F = A + B$	$x = \overline{F} = \overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

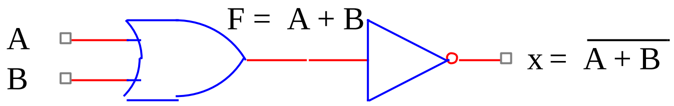
Bảng 5. Bảng thực trị của cổng NOR

- Cổng NOR là mạch số thực hiện phép toán NOR, có 2 ngõ nhập và 1 ngõ xuất và có ký hiệu cổng như sau:



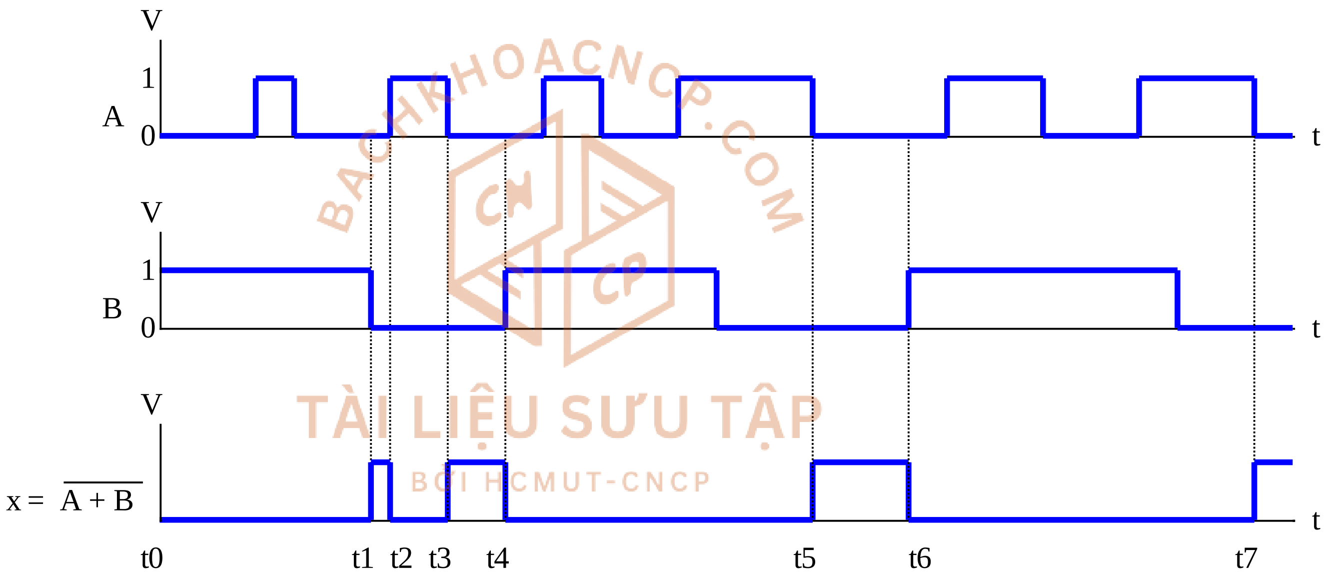
Hình 17. Ký hiệu cổng NOR

- Về nguyên tắc, thì cổng NOR có thể thay thế qua lại bằng hai cổng OR và NOT như hình sau:



Hình 18. Thay thế cổng NOR bằng hai cổng OR và NOT

- Giản đồ thời gian của hai biến A, B và hàm $x=\overline{A+B}$ được biểu diễn như hình sau:



Hình 19. Giản đồ thời gian của biến A, B và hàm $x=\overline{A+B}$

- Để vẽ được giản đồ thời gian của $x = \overline{A+B}$, ta chỉ cần quan tâm đến những thời điểm A và B đồng thời bằng 0 ⇒ x = 1, trong trường hợp này là các thời điểm t1→t2, t3→t4, t5→t6, t7→..., các trường hợp khác x đều bằng 0. Và chúng ta cũng thấy là giản đồ thời gian của hàm NOR/cổng NOR là nghịch đảo của hàm OR/cổng OR.

- Phép toán NOR và cổng NOR được ứng dụng rất nhiều, nên nó trở thành một phép toán và cổng cơ bản. Và người ta cũng đã chứng minh được là chỉ dùng duy nhất một phép toán NOR để xây dựng một hàm logic. Hay ở góc độ kỹ thuật, ta xây dựng được một mạch logic bất kỳ với một loại cổng duy nhất là cổng NOR.

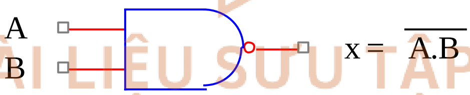
1.7.2- Phép toán NAND và cổng NAND

- Phép toán NAND là hàm hai biến x = f(A, B).
- Ký hiệu đại số: $x = \overline{A.B}$ (đọc là x bằng A nand B).
- Đây là phép toán dựa trên hai phép toán là NOT và AND, nó hoạt động theo bảng thực trị sau:

B	A	F = A.B	$x = \overline{F} = \overline{A.B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

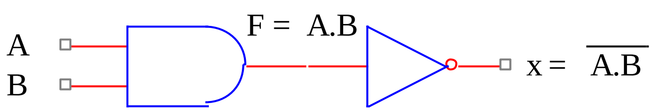
Bảng 6. Bảng thực trị của cổng NAND

- Cổng NAND là mạch số thực hiện phép toán NAND, có 2 ngõ nhập và 1 ngõ xuất và có ký hiệu cổng như sau:



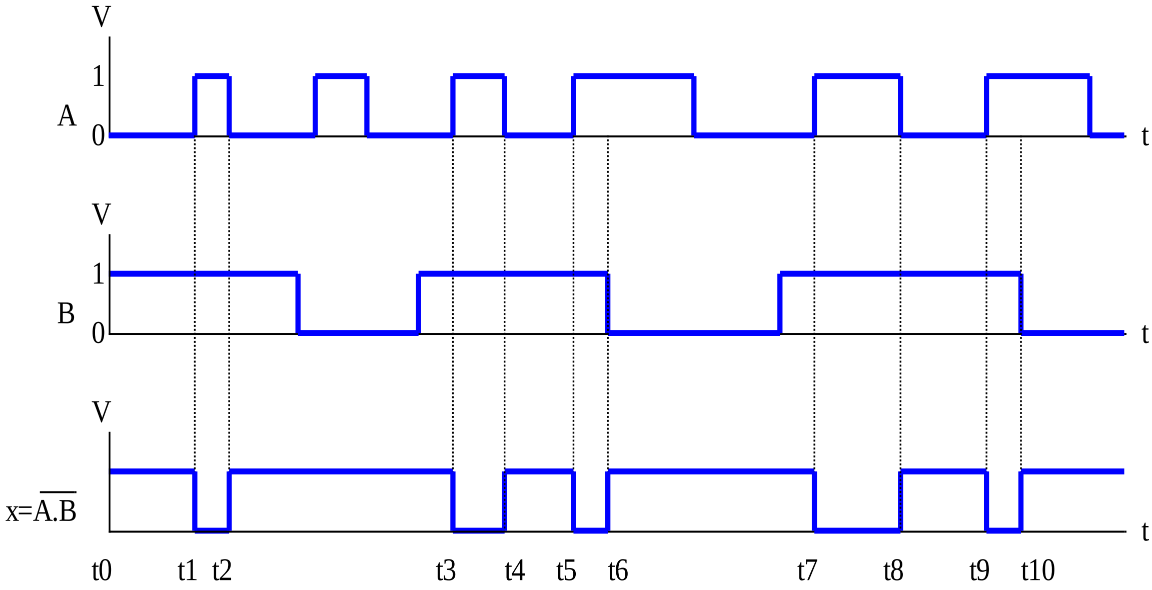
Hình 20. Ký hiệu cổng NAND

- Về nguyên tắc, thì cổng NAND có thể thay thế qua lại bằng hai cổng AND và NOT như hình sau:



Hình 21. Thay thế cổng NAND bằng hai cổng AND và NOT

- Giản đồ thời gian của hai biến A, B và hàm $x = \overline{A+B}$ được biểu diễn như hình sau:



Hình 22. Giản đồ thời gian của biến A, B và phép toán x= $\overline{A.B}$

- Để vẽ được giản đồ thời gian của x = $\overline{A.B}$, ta chỉ cần quan tâm đến những thời điểm A và B đồng thời bằng 0 ⇒ x = 1, trong trường hợp này là các thời điểm t1→t2, t3→t4, t5→t6, t7→t8, t9→t10, các trường hợp khác x đều bằng 1. Và chúng ta cũng thấy là giản đồ thời gian của hàm NAND/cổng NAND là nghịch đảo của hàm AND/cổng AND.

- Phép toán NAND và cổng NAND được ứng dụng rất nhiều, nên nó trở thành một phép toán và cổng cơ bản. Và người ta cũng đã chứng minh được là chỉ dùng duy nhất một phép toán NAND để xây dựng một hàm logic. Hay ở góc độ kỹ thuật, ta xây dựng được một mạch logic bất kỳ với một loại cổng duy nhất là cổng NAND.

1.7.3- Phép toán EX-OR và cổng EX-OR

- Phép toán EX-OR là hàm hai biến x = f(A, B).

- Ký hiệu đại số: x = $A \oplus B$ (đọc là x bằng A ex-or B).

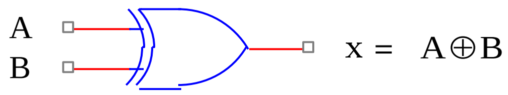
- Đây là phép toán dựa trên ba phép toán là NOT, AND và OR, nó hoạt động theo bảng thực trị sau:

B	$\overline{B}$	A	$\overline{A}$	F = $A.\overline{B}$	G = $\overline{A}.B$	x= F+G = $A.\overline{B} + \overline{A}.B = A \oplus B$
0	1	0	1	0	0	0
0	1	1	0	1	0	1

1	0	0	1	0	1	1
1	0	1	0	0	0	0

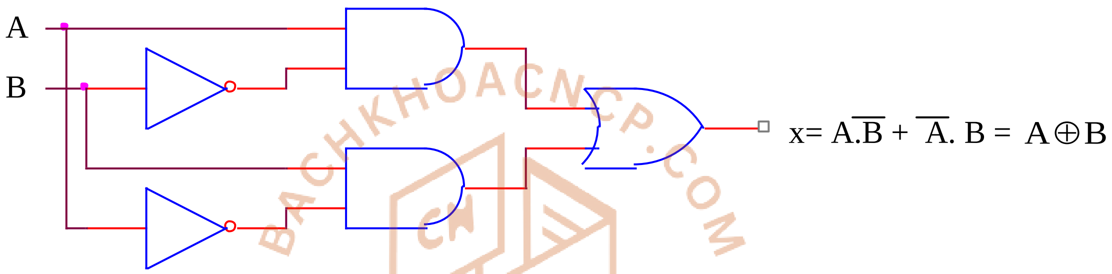
Bảng 7. Bảng thực trị của cổng EX-OR

- Cổng EX-OR là mạch số thực hiện phép toán EX-OR , có 2 ngõ nhập và 1 ngõ xuất và có ký hiệu cổng như sau:



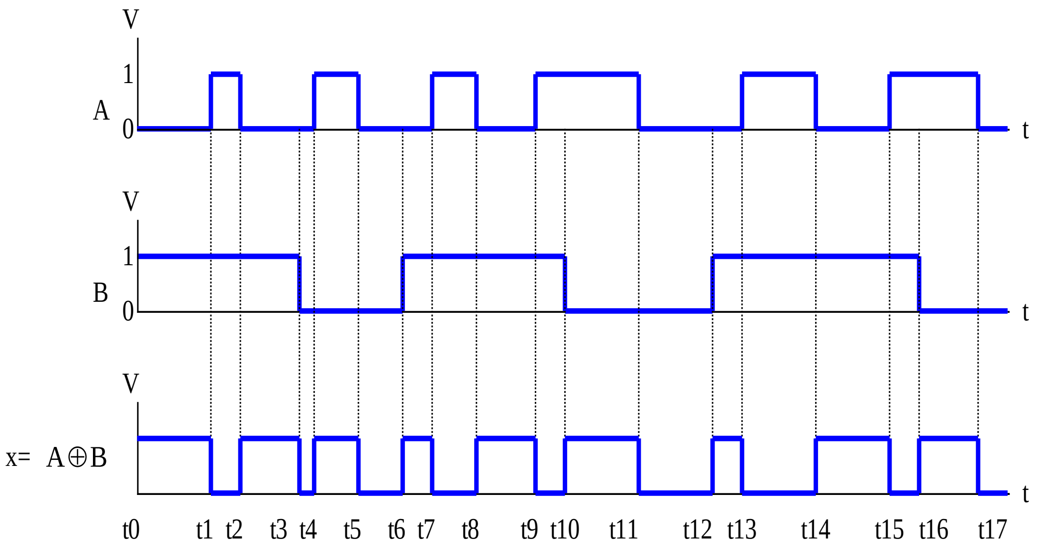
Hình 23. Ký hiệu cổng EX-OR

- Về nguyên tắc, thì cổng EX-OR có thể thay thế qua lại bằng ba loại cổng AND, OR và NOT như hình sau:



Hình 24. Thay thế cổng EX-OR bằng ba loại cổng AND, OR và NOT

- Giản đồ thời gian của hai biến A, B và phép toán x= A⊕B được biểu diễn như hình sau:



Hình 25. Giản đồ thời gian của biến A, B và hàm x= A⊕B

➢ Để vẽ được giản đồ thời gian của x = $\overline{A \oplus B}$, ta chỉ cần quan tâm đến những thời điểm A và B đồng thời bằng 0 hay đồng thời bằng 1 ⇒ x = 0, trong trường hợp này là các thời điểm t1→t2, t3→t4, t5→t6, t7→t8, t9→t10, t11→t12, t13→t14, t15→t16, t17→..., các trường hợp khác x đều bằng 1.

1.7.4- Phép toán EX-NOR và cổng EX-NOR

➢ Phép toán EX-NOR là hàm hai biến x = f(A, B).

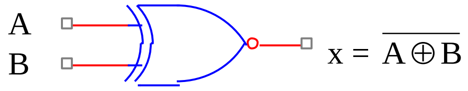
➢ Ký hiệu đại số: x = $\overline{A \oplus B}$ (đọc là x bằng A ex-nor B).

➢ Đây là phép toán dựa trên ba phép toán là NOT, AND và OR, nó hoạt động theo bảng thực trị sau:

B	$\overline{B}$	A	$\overline{A}$	F = $\overline{A}.\overline{B}$	G = A.B	x= F+G = $\overline{A}.\overline{B}$+A.B= $\overline{A \oplus B}$
0	1	0	1	1	0	1
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	1	1

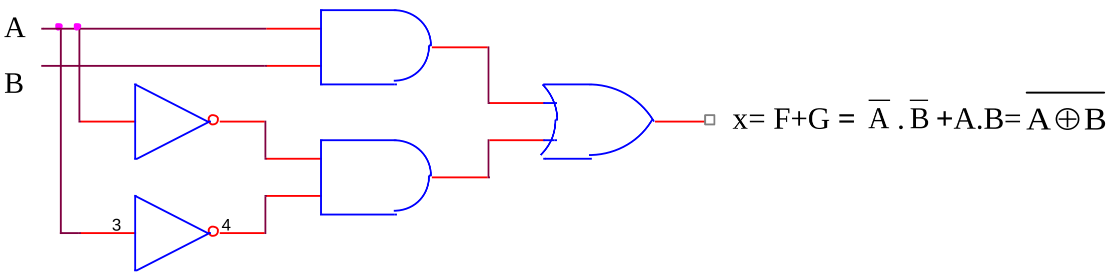
Bảng 8. Bảng thực trị của cổng EX-NOR

➢ Cổng EX-NOR là mạch số thực hiện phép toán EX-NOR, có 2 ngõ nhập và 1 ngõ xuất và có ký hiệu cổng như sau:



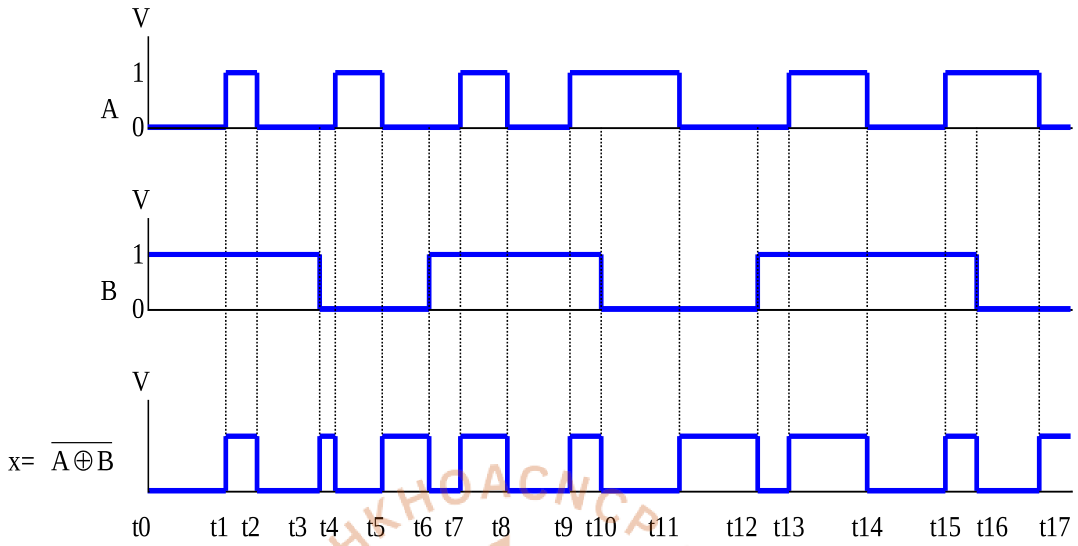
Hình 26. Ký hiệu cổng EX-NOR

➢ Về nguyên tắc, thì cổng EX-NOR có thể có thể thay thế qua lại bằng ba loại cổng AND, OR và NOT như hình sau:



Hình 27. Thay thế cổng EX-NOR bằng ba loại cổng AND, OR và NOT

- Giản đồ thời gian của hai biến A, B và phép toán $x = \overline{A \oplus B}$ được biểu diễn như hình sau:



Hình 28. Giản đồ thời gian của biến A, B và hàm $x = \overline{A \oplus B}$

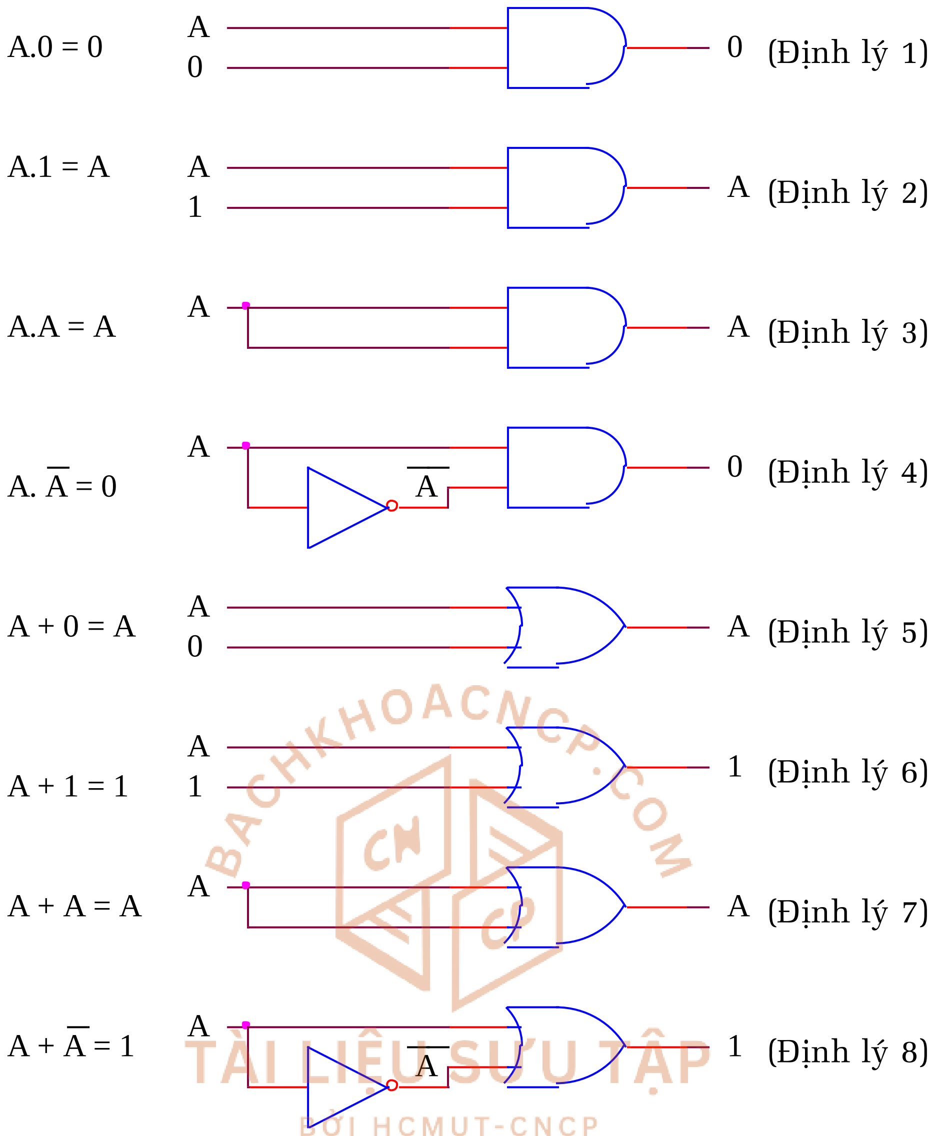
- Để vẽ được giản đồ thời gian của $x = \overline{A \oplus B}$, ta chỉ cần quan tâm đến những thời điểm A và B đồng thời bằng 0 hay đồng thời bằng 1 $\Rightarrow$ x = 1, trong trường hợp này là các thời điểm t1→t2, t3→t4, t5→t6, t7→t8, t9→t10, t11→t12, t13→t14, t15→t16, t17→..., các trường hợp khác x đều bằng 0. Và chúng ta cũng thấy là giản đồ thời gian của phép toán EX-NOR/cổng EX-NOR là nghịch đảo của phép toán EX-OR/cổng EX-OR .

1.8- Các Định Lý Của Đại Số Logic

Từ những ba phép toán cơ bản NOT, OR, AND, người ta xây dựng một số định lý nhằm đơn giản biểu thức logic hay mạch logic. Các định lý này sẽ được đi kèm với sơ đồ mạch logic minh họa.

1.8.1- Các định lý cho biểu thức 1 biến hay 1 biến kết hợp với hằng

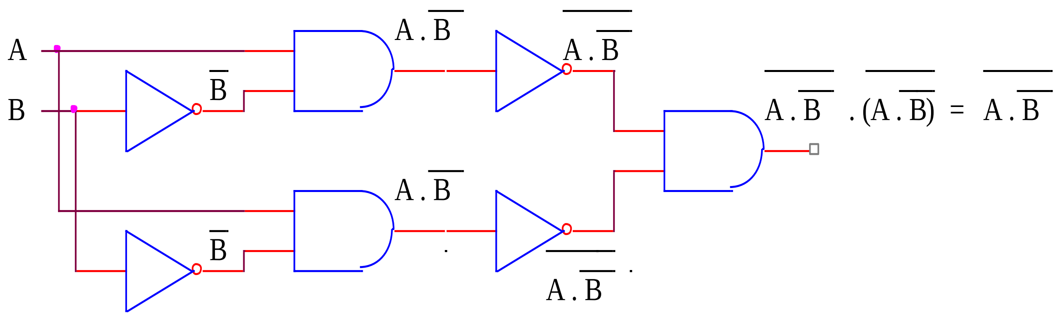
- Trước hết, ta xét các định lý cho 1 biến hay 1 biến kết hợp với hằng 0 hay hằng 1 như các định lý từ 1→8 ở dưới:



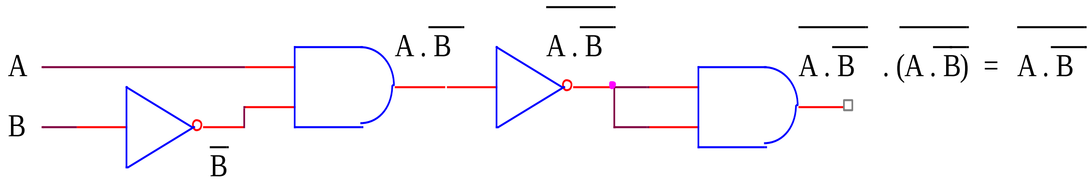
Hình 29. Các định lý cho 1 biến, hay 1 biến kết hợp với 1 hằng

- Tất cả định lý trên đều có thể chứng minh dựa trên bảng thực trị, nhưng riêng với phép toán AND, ta có thể xem nó như là phép nhân cho dễ nhớ. Nếu nhân với 0 thì sẽ bằng 0, còn nhân với 1 sẽ bằng chính nó.

- Ta cũng phải lưu ý đến việc mở rộng phạm vi áp dụng các định lý trên bằng cách dùng phương pháp thay thế một biểu thức logic thành một biến. Như trường hợp định lý 3, với biểu thức $\overline{A.\overline{B}}.(\overline{A.\overline{B}})$, thì ta đặt biến phụ F = $\overline{A.\overline{B}}$ thì biểu thức logic trên sẽ trở thành là F.F = F = $\overline{A.\overline{B}}$ ⇒ $\overline{A.\overline{B}}.(\overline{A.\overline{B}})$ = $\overline{A.\overline{B}}$ như hình minh họa dưới.



Hay ta dùng mạch tương đương sau

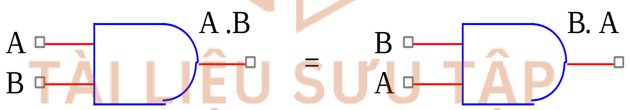


Hình 30. Mở rộng phạm vi áp dụng cho một biểu thức

1.8.2- Các định lý về luật giao hoán và kết hợp

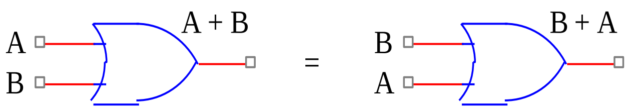
- Ta xét tiếp đến các định lý về luật giao hoán cho phép toán AND và OR. Với hai định lý này, ta thấy là thứ tự các toán hạng là không quan trọng, và ta dễ dàng chứng minh bằng bảng thực trị

A.B = B.A



(Định lý 9)

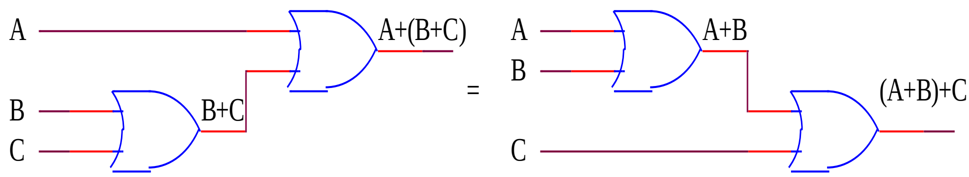
A+B = B+A

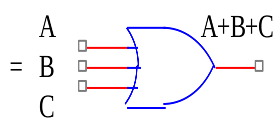


(Định lý 10)

Hình 31. Luật giao hoán cho phép toán AND và OR

- Ta xét tiếp định lý về luật kết hợp cho phép toán OR như hình vẽ dưới.





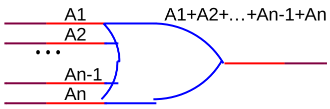
A+(B+C) = (A+B)+C = A+B+C

(Định lý 11)

Hình 32. Luật kết hợp cho phép toán OR

- Thứ tự thực hiện phép toán OR ở đây không quan trọng, Và như vậy ta có thể biểu diễn phép toán này theo nhiều cách như hình vẽ trên minh họa. Và ta cũng có nhận xét là có thể xây dựng cổng OR 3 ngõ nhập để tương ứng với cách biểu diễn A + B + C.

- Và ta có thể phát triển định lý 11 cho phép toán OR có N biến, ta biểu diễn nó như sau: $A_0 + A_1 + A_2 + A_3 + ... + A_{N-2} + A_{N-1}$. Khi thể hiện nó bằng cổng logic, về mặt lý thuyết, ta sẽ thể hiện biểu thức này bằng cổng OR có N ngõ nhập.



Hình 33. Ký hiệu cổng OR có N ngõ nhập

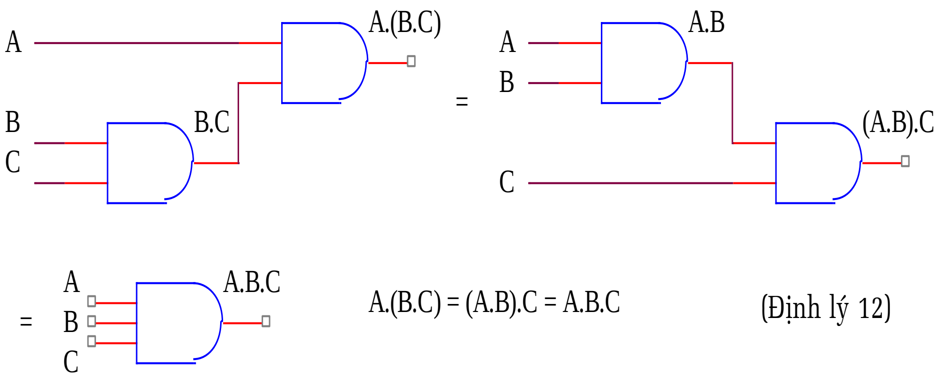
- Lập bảng thực trị cho phép toán OR có N ngõ nhập, ta thấy là giá trị của $x= A_{n-1} + A_{n-2} +... + A_2 + A_1 + A_0$ chỉ bằng 0 với một trường hợp duy nhất là $(A_{n-1}, A_{n-2},..., A_2, A_1, A_0) = (0, 0, ..., 0, 0, 0)$ còn mọi trường hợp khác thì x = 1.

A_{n-1}	A_{n-2}	...	A_2	A_1	A_0	$x= A_{n-1} + A_{n-2} +... + A_2 + A_1 + A_0$
0	0	...	0	0	0	0
0	0	...	0	0	1	1
. . .	. . .		. . .	. . .	. . .	. . .
1	1	...	1	1	0	1
1	1	...	1	1	1	1

Bảng 9. Bảng thực trị cho phép toán OR có N ngõ nhập

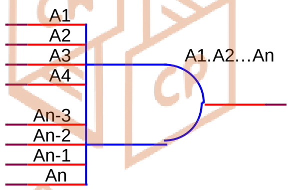
- Về mặt kỹ thuật. các nhà sản xuất cũng đã sản xuất các cổng OR với số ngõ nhập từ 2 ngõ lên đến 8 ngõ. Nhưng do những loại cổng OR có từ 3 ngõ nhập trở lên thì khá hiếm trên thị trường, nên khi thi công ta vẫn phải dùng cổng OR hai ngõ nhập thay thế.

- Ta xét tiếp các định lý về luật kết hợp cho phép toán AND. Thứ tự thực hiện phép toán AND ở đây không quan trọng, Và như vậy ta có thể biểu diễn phép toán này theo nhiều cách như hình vẽ dưới. Và ta cũng có nhận xét là có thể xây dựng cổng AND 3 ngõ nhập để tương ứng với cách biểu diễn A.B.C



Hình 34. Luật kết hợp cho phép toán AND.

- Và ta có thể phát triển định lý 12 cho phép toán AND có N biến, ta biểu diễn nó như sau: $A_0.A_1.A_2.A_3$ . ... . $A_{N-2}.A_{N-1}$. Khi thể hiện nó bằng cổng logic, về mặt lý thuyết, ta sẽ thể hiện biểu thức này bằng cổng AND có N ngõ nhập.



Hình 35. Ký hiệu cổng AND có N ngõ nhập

- Lập bảng thực trị cho phép toán AND có N ngõ nhập, ta thấy là giá trị của x= A_{n-1} . A_{n-2} .... . A_2 . A_1 . A_0 chỉ bằng 1 với một trường hợp duy nhất là (A_{n-1}, A_{n-2},..., A_2, A_1, A_0) = (1, 1, ..., 1, 1, 1) còn mọi trường hợp khác thì x = 0.

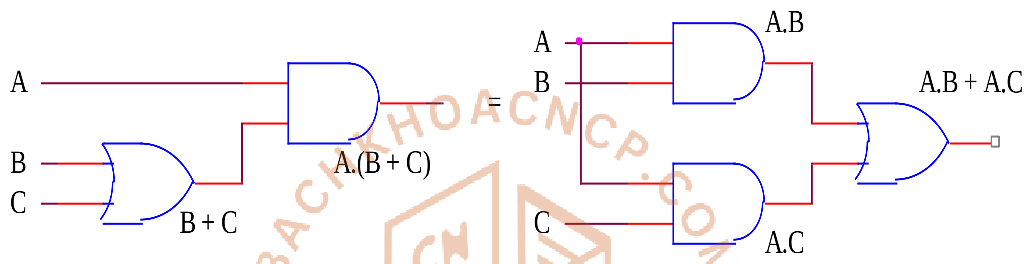
A_{n-1}	A_{n-2}	...	A_2	A_1	A_0	x= A_{n-1} . A_{n-1} .. . A_2 . A_1 . A_0
0	0	...	0	0	0	0
0	0	...	0	0	1	0
. . .	. . .		. . .	. . .	. . .	. . .
1	1	...	1	1	0	0
1	1	...	1	1	1	1

Bảng 10. Bảng thực trị cho phép toán AND có N ngõ nhập

- Về mặt kỹ thuật. các nhà sản xuất cũng đã sản xuất các cổng AND với số ngõ nhập từ 2 ngõ lên đến 8 ngõ. Nhưng do những loại cổng AND có từ 3 ngõ nhập trở lên thì khá hiếm trên thị trường, nên khi thi công ta vẫn phải dùng cổng AND hai ngõ nhập thay thế.
- Để chứng minh các định lý về luật kết hợp của phép toán OR và AND này ta dùng bảng thực trị để chứng minh.

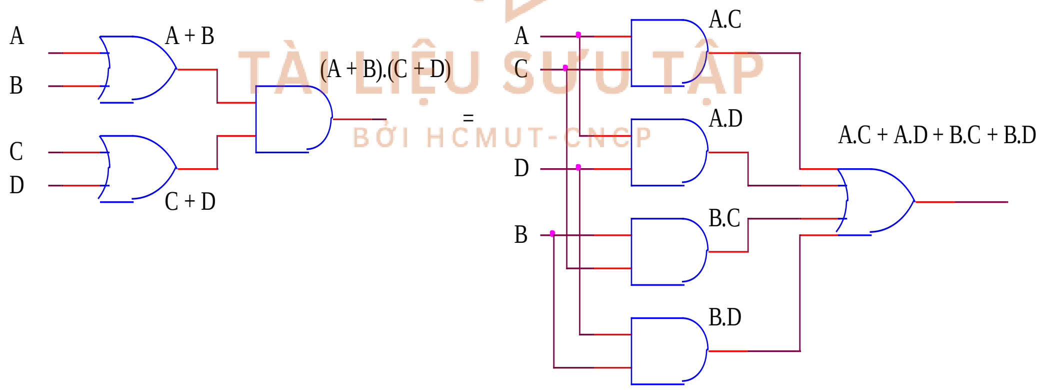
1.8.3- Các định lý về luật phân phối giữa phép toán OR và phép toán AND

(Định lý 13.a) A.(B+C) = A.B + A.C



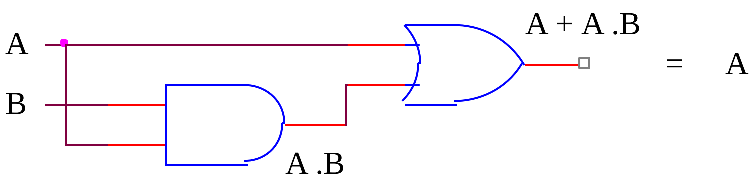
Hình 36. Minh họa định lý 13.a

(Định lý 13.b) (A+B).(C+D) = A.C + A.C + B.C + B.D



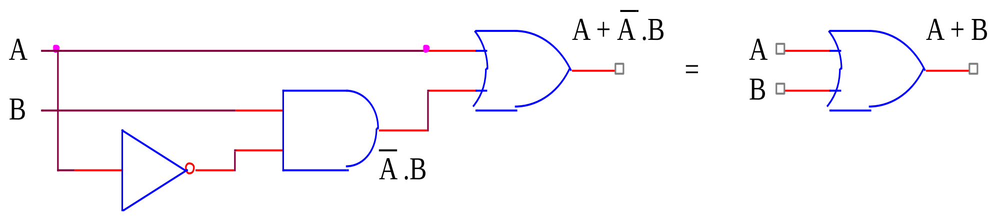
Hình 37. Minh họa định lý 13.b

(Định lý 13.c) A + A.B = A



Hình 38. Minh họa định lý 13.c

(Định lý 13.d) A + $\overline{A}$.B = A + B



Hình 39. Minh họa định lý 13.d

- Các định lý (13.a) và (13.b) có thể chứng minh bằng cách dùng bảng thực trị. Còn để dễ nhớ, ta xem luật này giống như cách ta khai triển biểu thức và đặt thừa số chung trong đại số thông thường.

- Các định lý (13.c) và (13,d) là đặc thù riêng của đại số logic, ta có thể dùng bảng thực trị để chứng minh hay áp dụng các định lý trước đó để chứng minh.

- Ta dùng các định lý trước để chứng minh định lý (13.c):

  A + A.B = A.1 + A.B

  áp dụng định lý (2) A = A.1

  A.1 + A.B = A(1 + B)

  áp dụng định lý (13.a) A.B + A.C = A.(B+C)

  A(1 + B) = A . 1

  áp dụng định lý (6) 1 + B = 1.

  A.1 = A

  áp dụng định lý (2) A.1= A

  Vậy: A + A.B = A định lý đã được chứng minh.

- Ta dùng các định lý trước để chứng minh định lý (13.d):

  $A + \overline{A}.B = A.1 + \overline{A}.B$

  áp dụng định lý (2) $A = A.1$

  $A.1 + \overline{A}.B = A.(B+\overline{B}) + \overline{A}.B$

  áp dụng định lý (8) $1 = B + \overline{B}$

  $A.(B+\overline{B}) + \overline{A}.B = A.B + A.\overline{B} + \overline{A}.B$

  áp dụng định lý (13.a) $A.(B+\overline{B})=A.B + A.\overline{B}$

  $A.B + A.\overline{B} + \overline{A}.B = A.B + A.B + A.\overline{B} + \overline{A}.B$

  áp dụng định lý (3) A.B=(A.B).(A.B)

  $A.B + A.B + A.\overline{B} + \overline{A}.B = A.(B+\overline{B}) + B.(A+\overline{A})$

  áp dụng định lý (13.a) $A.B+A.\overline{B}=A.(B+\overline{B})$ và $A.B+\overline{A}.B=B(A+\overline{A})$

  $A.(B+\overline{B}) + B.(A+\overline{A}) = A.1 + B.1$

  áp dụng định lý (8) $B + \overline{B} = 1$ và $A+\overline{A} = 1$

  $A.1 + B.1 = A + B$

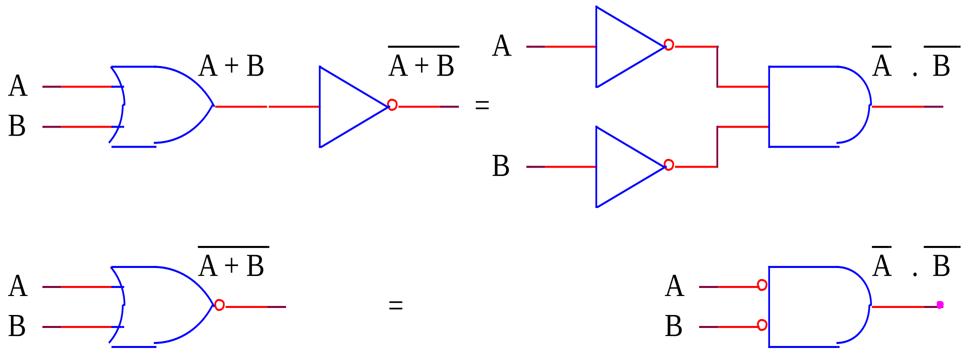
áp dụng định lý (2) A.1= A và B.1=B

Vậy: $A + \overline{A}.B = A + B$ định lý đã được chứng minh.

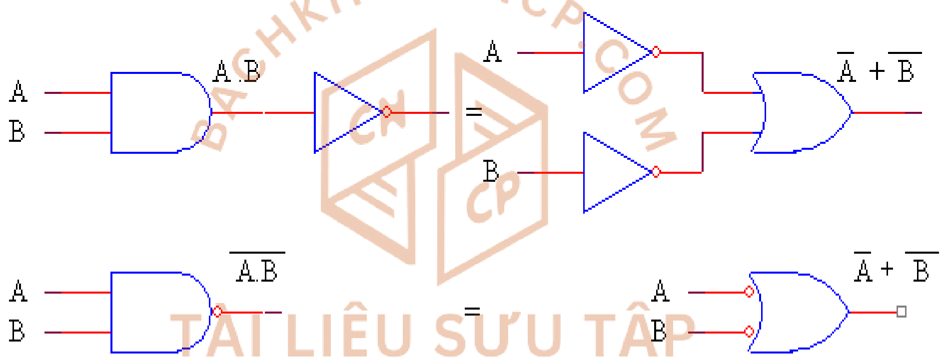
1.8.4- Các định lý DeMorgan

(Định lý 14) $\overline{A+B} = \overline{A}.\overline{B}$

(Định lý 15) $\overline{A.B} = \overline{A} + \overline{B}$



Hình 40. Minh họa cho định lý 14

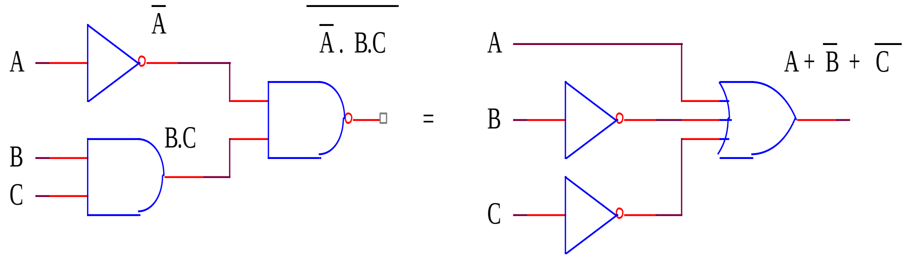


Hình 41. Minh họa cho định lý 15

- Ta có thể chứng minh hai định lý trên bằng cách lập bảng thực trị. Đây là hai định lý rất quan trọng trong việc rút gọn các biểu thức có dạng phủ định của một tổng hay dạng phủ định của một tích.

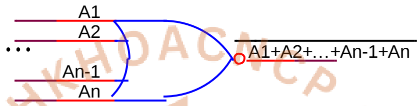
- Nhận xét: định lý số 14 chính là phép toán NOR/cổng NOR, định lý 15 chính là phép toán NAND/cổng NAND. Với định lý DeMorgan, ta có một cách biểu diễn khác cho cho hai cổng trên như vế phải của các hình trên.

- Ta cũng phải lưu ý đến việc mở rộng phạm vi áp dụng các định lý trên bằng cách dùng phương pháp thay thế một biểu thức logic thành một biến. Như trường hợp định lý 14, với biểu thức $\overline{\overline{A}.B.C}$, thì ta đặt biến phụ F = $\overline{A}$ và G = B.C thì biểu thức logic trên sẽ trở thành là $\overline{F.G} = \overline{F} + \overline{G} = \overline{\overline{A}} + \overline{B.C}$ $= A + \overline{B} + \overline{C} \Rightarrow \overline{\overline{A}.B.C} = A + \overline{B} + \overline{C}$ như hình minh họa dưới.



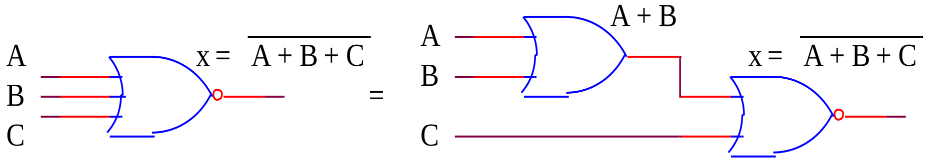
Hình 42. Mở rộng phạm vi áp dụng cho một biểu thức

- Và ta có thể phát triển định lý 14 cho phép toán NOR có N biến, ta biểu diễn nó như sau: $A_1 + A_2 + A_3 + ... + A_{N-1} + A_N$. Khi thể hiện nó bằng cổng logic, về mặt lý thuyết, ta sẽ thể hiện biểu thức này bằng cổng NOR có N ngõ nhập.



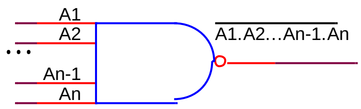
Hình 43. Ký hiệu cổng NOR có N ngõ nhập

- Về mặt kỹ thuật, các nhà sản xuất cũng đã sản xuất các cổng NOR với số ngõ nhập từ 2 ngõ lên đến 8 ngõ. Nhưng do những loại cổng OR có từ 3 ngõ nhập trở lên thì khá hiếm trên thị trường, nên khi thi công ta vẫn phải dùng cổng NOR hai ngõ nhập và cổng OR hai ngõ nhập để thay thế. Như trường hợp cổng NOR 3 ngõ nhập dưới, phải thay thế bằng 1 cổng OR hai ngõ nhập và 1 cổng NOR 2 ngõ nhập.



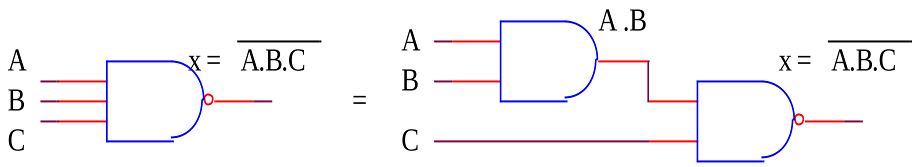
Hình 44. Thay thế cổng NOR 3 ngõ nhập

- Và ta có thể phát triển định lý 15 cho phép toán NAND có N biến, ta biểu diễn nó như sau: $A_1 + A_2 + A_3 + ... + A_{N-1} + A_N$. Khi thể hiện nó bằng cổng logic, về mặt lý thuyết, ta sẽ thể hiện biểu thức này bằng cổng NAND có N ngõ nhập.



Hình 45. Ký hiệu cổng NAND có N ngõ nhập

- Về mặt kỹ thuật. các nhà sản xuất cũng đã sản xuất các cổng NAND với số ngõ nhập từ 2 ngõ lên đến 8 ngõ. Nhưng do những loại cổng NAND có từ 3 ngõ nhập trở lên thì khá hiếm trên thị trường, nên khi thi công ta vẫn phải dùng cổng NAND hai ngõ nhập và cổng AND hai ngõ nhập để thay thế. Như trường hợp cổng NAND 3 ngõ nhập dưới, phải thay thế bằng 1 cổng AND hai ngõ nhập và 1 cổng NAND 2 ngõ nhập.

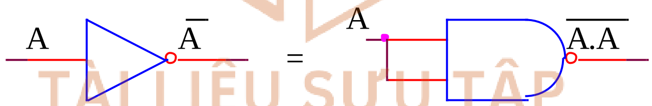


Hình 46. Thay thế cổng NAND 3 ngõ nhập

1.8.5- Tính đa dụng của phép toán NOR/ cổng NOR và phép toán NAND/ cổng NAND

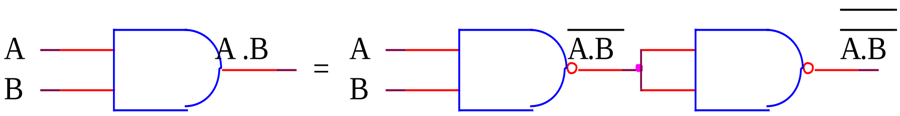
- Như trên đã nói, có thể dùng duy nhất một loại cổng NOR hay NAND để thiết kế mạch logic. Ta sẽ chứng minh bằng các định lý đã nêu ở phần trước.

- Thay thế cổng NOT bằng cổng NAND: $\overline{A} = \overline{A.A}$



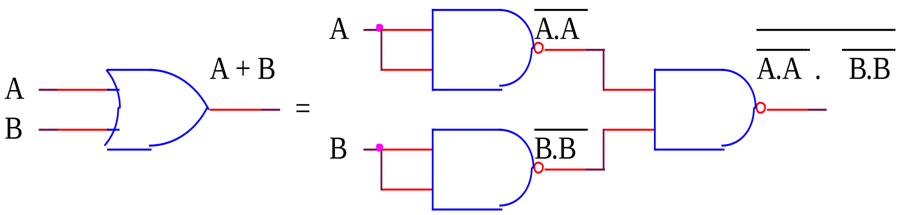
Hình 47. Thay thế một cổng NOT bằng một cổng NAND

- Thay thế cổng AND bằng cổng NAND: $A.B = \overline{\overline{A.B}}$



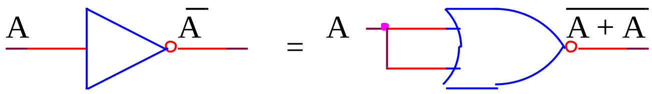
Hình 48. Thay thế một cổng AND bằng hai cổng NAND

- Thay thế cổng OR bằng cổng NAND: $A + B = \overline{\overline{A.A}.\overline{B.B}}$



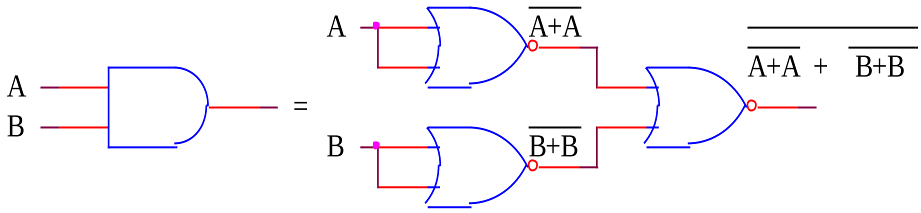
Hình 49. Thay thế một cổng OR bằng ba cổng NAND

- Thay thế cổng NOT bằng cổng NOR: $\overline{A} = \overline{A+A}$



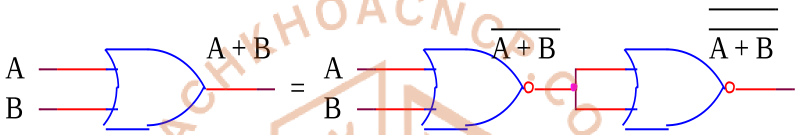
Hình 50. Thay thế một cổng NOT bằng một cổng NOR

- Thay thế cổng AND bằng cổng NOR: $A.B = \overline{\overline{A+A}+\overline{B+B}}$



Hình 51. Thay thế một cổng AND bằng ba cổng NOR

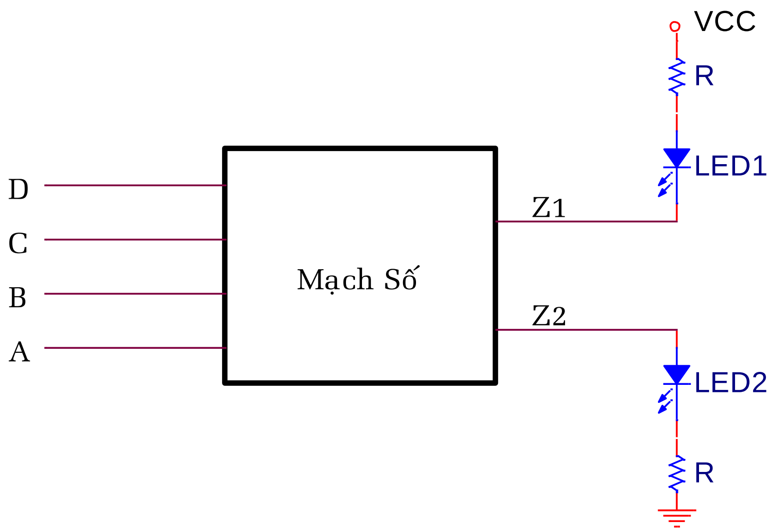
- Thay thế cổng OR bằng cổng NOR: $A + B = \overline{\overline{A+B}}$



Hình 52. Thay thế một cổng OR bằng hai cổng NOR

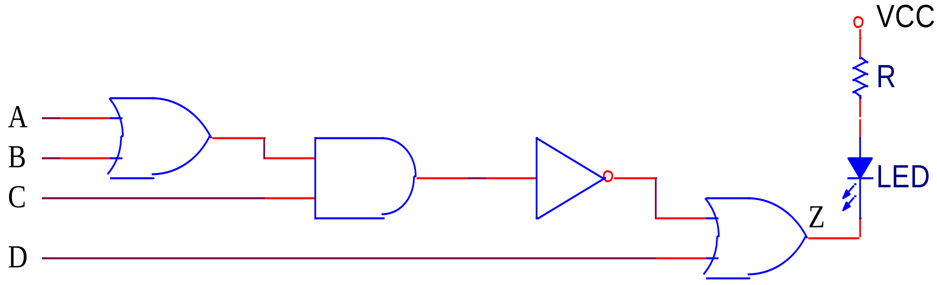
1.8.6- Mức logic tích cực

- Cho một mạch tổ hợp có sơ đồ khối như hình vẽ, đèn LED1 sẽ sáng khi $Z_1 = 0$ và đèn LED2 sẽ sáng khi $Z_2 = 1$. Trong trường hợp này ta nói, Z_1 là ngõ xuất tích cực mức thấp và Z_2 là ngõ xuất tích cực mức cao.



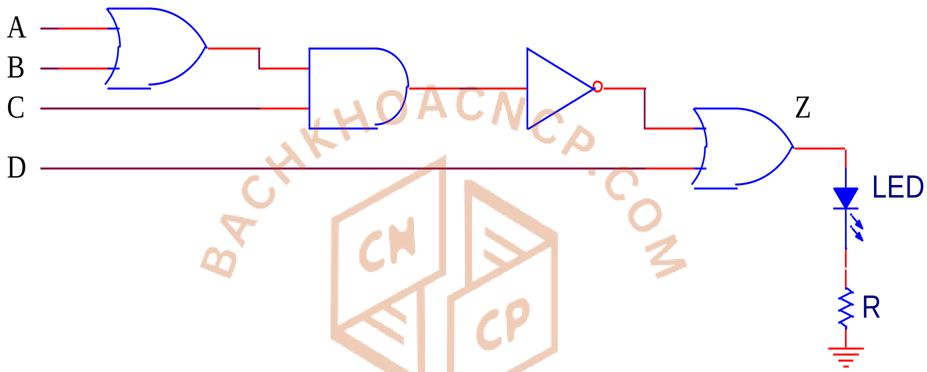
Hình 53. Sơ đồ khối cho khái niệm mức logic tích cực.

➢ Để minh họa rõ khái niệm mức logic tích cực, ta xét một trường hợp cụ thể sau:



Hình 54. Z là ngõ xuất tích cực mức thấp.

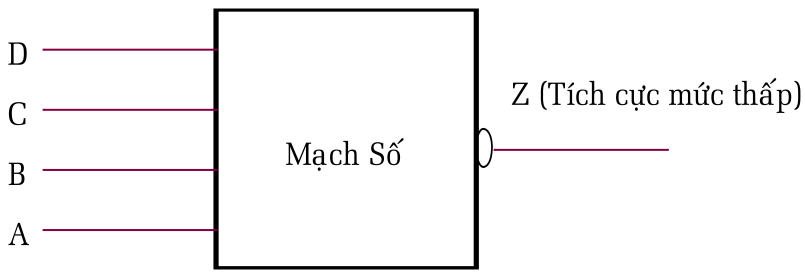
➢ Cũng lấy mạch tổ hợp này, nhưng lần này nối với Led theo cách khác.



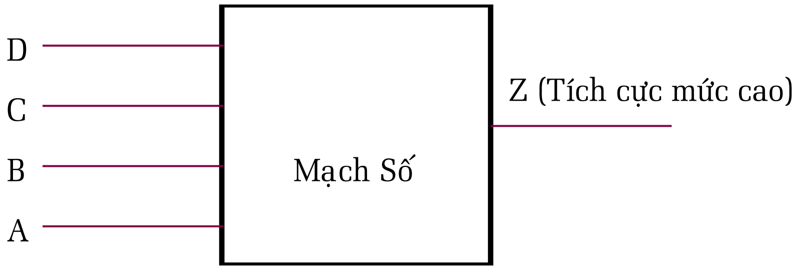
Hình 55. Z là ngõ xuất tích cực mức cao.

➢ Với mạch tổ hợp không thay đổi, chỉ nối với đèn LED theo hai cách khác nhau, ta có khái niệm ngõ xuất tích cực mức thấp và tích cực mức cao. Mạch trước, đèn sáng khi Z = 0, ta nói mạch này có ngõ xuất tích cực mức thấp. Mạch sau, đèn sáng khi Z = 1, ta nói mạch này có ngõ xuất tích cực mức cao. Còn bản chất của mạch tổ hợp không có gì thay đổi.

➢ Về mặt ký hiệu, thì sử dụng dấu tròn (°) để biểu thị tích cực mức thấp và bình thường để biểu thị tích cực mức cao.



Hình 56. Ký hiệu cho ngõ xuất tích cực mức thấp.



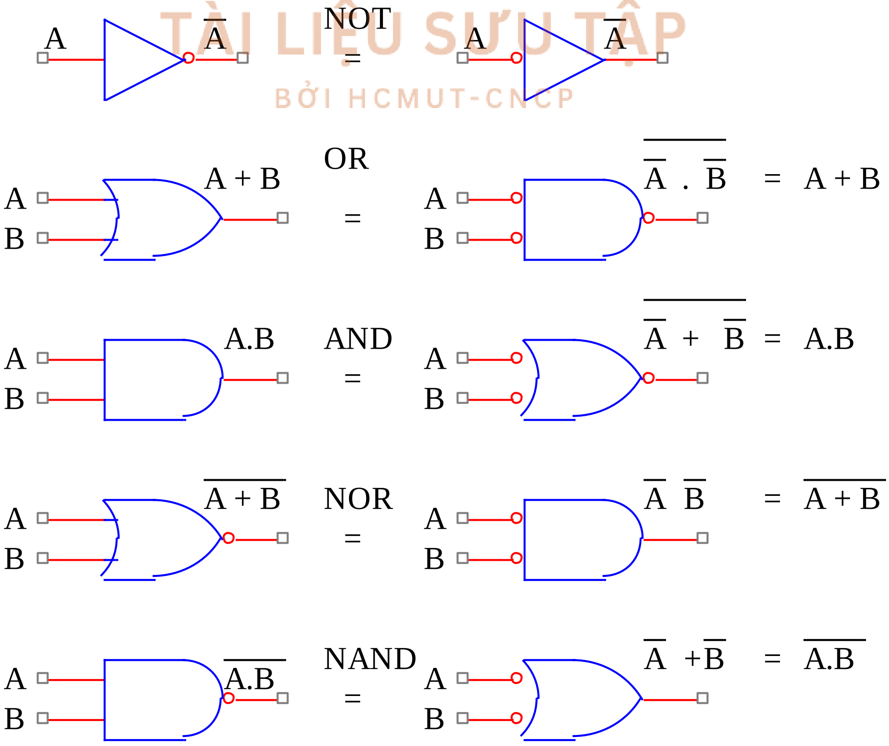
Hình 57. Ký hiệu cho ngõ xuất tích cực mức cao.

- Và khái niệm mức logic tích cực cũng áp dụng cho cả ngõ nhập, với ngõ nhập nào bằng 1 mà ngõ xuất tích cực thì ta nói ngõ nhập đó tích cực mức cao, ngõ nhập bằng 0 mà ngõ xuất tích cực thì ta nói ngõ nhập đó tích cực mức thấp, và ký hiệu mức logic tích cực cũng như ngõ xuất.

- Khái niệm mức logic tích cực được sử dụng với mục đích là phân tích mạch bằng lý luận (không dùng bảng thực trị) để hiểu được cách hoạt động của mạch.

1.8.7- Ký hiệu logic chuẩn và ký hiệu logic thay thế

- Với khái niệm mức logic tích cực đã đề cập trên, ta sẽ dùng chúng để thay đổi ký hiệu 5 loại cổng NOT, OR, AND, NOR, NAND. Những ký hiệu đó được gọi là ký hiệu thay thế. Và chúng được dùng khi phân tích mạch bằng lý luận.

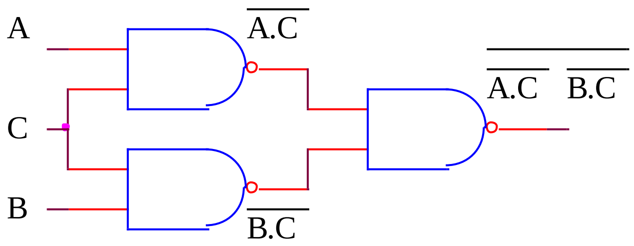


Hình 58. Ký hiệu chuẩn và các ký hiệu thay thế.

- ➢ Nhận xét 1: có thể mở rộng cho số ngõ nhập tới N.
- ➢ Nhận xét 2: toàn bộ ngõ nhập của ký hiệu chuẩn không có dấu tròn, còn toàn bộ ngõ nhập của ký hiệu thay thế đều có dấu tròn.

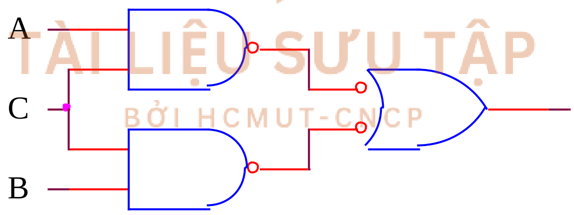
1.8.8- Ứng dụng của việc sử dụng ký hiệu logic thay thế

- ➢ Cho một sơ đồ mạch như sau:



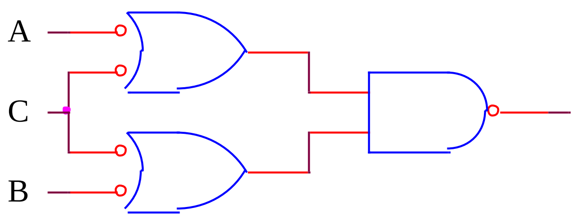
Hình 59. Mạch dùng toàn cổng NAND

- ➢ Rất khó nhận xét mạch hoạt động như thế nào, chỉ có thể nhận xét được sau khi lập xong bảng thực trị.
- ➢ Ta sẽ thay đổi ký hiệu logic chuẩn bằng ký hiệu logic thay thế để có thể để nhận xét mạch theo hai cách sau
- ➢ Cách 1:



Hình 60. Thay cổng NAND cuối bằng ký hiệu thay thế.

- ➢ Nhìn vào hình vẽ ta nói, ngõ xuất của mạch bằng 1 khi (A=1 và C=1) hay (B=1 và C=1).
- ➢ Cách 2:



Hình 61. Thay thế hai cổng NAND đầu bằng ký hiệu thay thế.

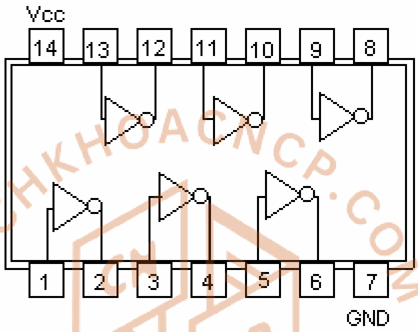
➢ Nhìn vào hình vẽ ta nói, ngõ xuất của mạch bằng 0 khi (A=0 hoặc B=0) và (C=0).

➢ Lưu ý, khi dùng các ký hiệu thay thế, phải bảo đảm nguyên tắc khi nối giữa các cổng thì dấu tròn (°) phải nối với dấu tròn (°), không dấu nối với không dấu.

➢ Cách thay thế thứ nhất, dùng khi ngõ xuất là tích cực mức cao. Cách thay thế thứ hai dùng khi ngõ xuất là tích cực mức thấp.

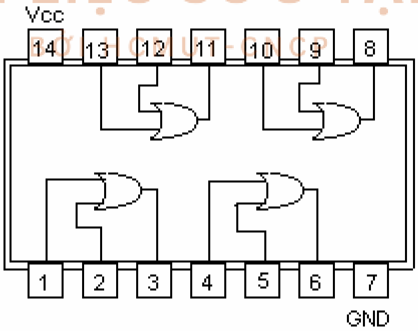
1.9- Các Vi Mạch Chứa Các Cổng Logic

1.9.1- Vi mạch chứa cổng NOT



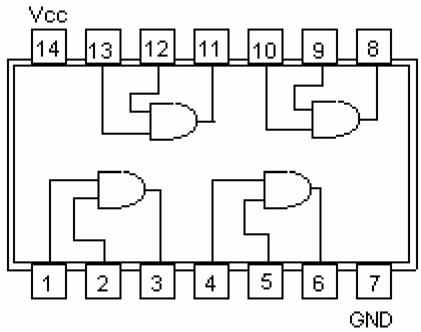
Hình 62. IC 7404 chứa sáu cổng NOT

1.9.2- Vi mạch chứa cổng OR



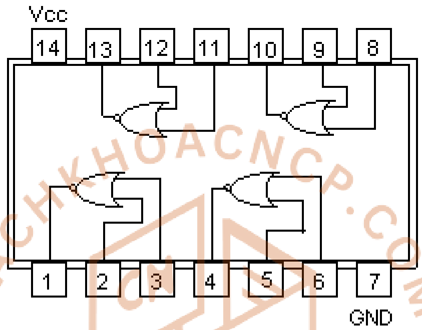
Hình 63. IC 7432 chứa 4 cổng OR

1.9.3- Vi mạch chứa cổng AND



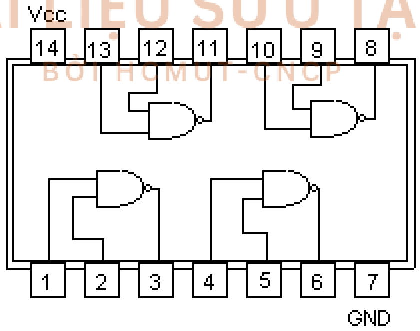
Hình 64. IC 7408 chứa 4 cổng NAND

1.9.4- Vi mạch chứa cổng NOR



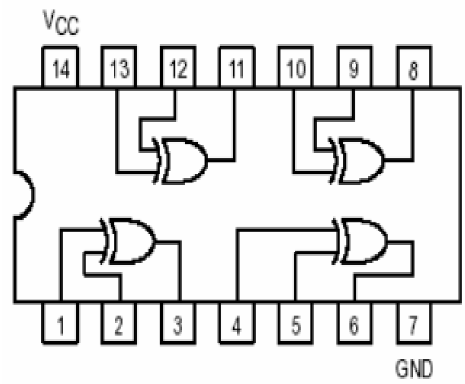
Hình 65. IC 7402 chứa 4 cổng NOR

1.9.5- Vi mạch chứa cổng NAND



Hình 66. IC 7400 chứa 4 cổng NAND

1.9.6- Vi mạch chứa cổng EX-OR



Hình 67. IC 7486 chứa 4 cổng EX-OR

1.10- Biểu Diễn Hàm Logic Theo Các Dạng Chuẩn

1.10.1- Dạng chuẩn tuyển và dạng chuẩn hội

- Để biểu diễn hàm logic bằng phương pháp đại số, có hai dạng chuẩn: ***dạng chuẩn tuyển*** (còn được gọi là dạng chuẩn giao hay tiếng Anh gọi là ***minterm*** hay ***sum of products***) và ***dạng chuẩn hội*** (tiếng Anh gọi là ***maxterm*** hay ***product of sums***).

- Các phép toán logic dùng ở đây là phép toán OR và phép toán AND, riêng phép toán NOT chỉ dùng cho 1 biến.

- Trong dạng chuẩn tuyển, mỗi biểu thức là một phép toán AND có N biến, các biến của phép toán AND có thể ở dạng bình thường hay qua phép toán NOT, tất cả các biểu thức đó sẽ là biến đầu vào của phép toán OR có N biến.

- Trong dạng chuẩn hội, mỗi biểu thức là một phép toán OR có N biến, các biến của phép toán OR có thể ở dạng bình thường hay qua phép toán NOT, tất cả các biểu thức đó sẽ là biến đầu vào của phép toán AND có N biến

- Ta sẽ đi lần lượt từ hàm logic có một biến đến N biến

1.10.2- Hàm logic 1 biến x = f(A)

- Lập bảng thực trị:

A	x = f(A)
0	x_0
1	x_1

Bảng 11. Bảng thực trị với $x_i \in \{0,1\}$

- Công thức đại số dạng chuẩn tuyển: $x = \overline{A}.x_0 + A.x_1$

- Công thức đại số dạng chuẩn hội: $x = (A + x_0).(\overline{A} + x_1)$

- Với hàm một biến, ta liệt kê tất cả trường hợp có thể xảy ra là $2^{2^1} = 2^2 = 4$ trường hợp.

- Trường hợp 1: $x_0 = x_1 = 0 \Rightarrow x = 0$ (Đây là hằng 0)

A	x = f(A)
0	0
1	0

Bảng 12. Bảng thực trị với $x_0 = x_1 = 0$

- Trường hợp 1-dạng chuẩn tuyển: $x = \overline{A}.x_0 + A.x_1 \Rightarrow x = \overline{A}.0 + A.0 = 0$

- Trường hợp 1-dạng chuẩn hội: $x = (A+x_0).(\overline{A}+x_1) \Rightarrow x=(\overline{A}+0).(A+0) = 0$

- Trường hợp 2: $x_0 = 0$ & $x_1 = 1 \Rightarrow x = A$

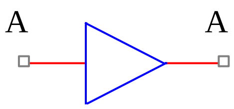
A	x = f(A)
0	0
1	1

Bảng 13. Bảng thực trị với $x_0 = 0$ & $x_1 = 1$

- Trường hợp 2-dạng chuẩn tuyển: $x = \overline{A}.x_0 + A.x_1 \Rightarrow x = \overline{A}.0 + A.1 = A$

- Trường hợp 2-dạng chuẩn hội: $x=(A+x_0).(\overline{A}+x_1) \Rightarrow x= (A+0).(\overline{A}+1) = A$

- Trong trường hợp x = A, về mặt lý thuyết, thì nối thẳng biến A vào mạch, không thông qua cổng nào. Trong thực tế thi công mạch, nếu có sự suy giảm điện thế của các ngõ xuất, ta thường thiết kế cổng đệm với ngõ xuất bằng ngõ nhập, để bảo đảm mức logic của tín hiệu. Hay trong trường hợp làm trung gian các loại cổng có công nghệ chế tạo khác nhau như giữa TTL và CMOS...



Bảng 14. Ký hiệu cổng đệm

- Trường hợp 3: $x_0 = 1$ & $x_1 = 0 \Rightarrow x = \overline{A}$ (Đây là cổng NOT)

A	x = f(A)
0	1
1	0

Bảng 15. Bảng thực trị với $x_0 = 1$ & $x_1 = 0$

- Trường hợp 3-dạng chuẩn tuyển: $x = \overline{A}.x_0 + A.x_1 \Rightarrow x = \overline{A}.1 + A.0 = \overline{A}$

- Trường hợp 3-dạng chuẩn hội: $x = (A+x_0).(\overline{A}+x_1) \Rightarrow x=(A+1).(\overline{A}+0) = \overline{A}$

- Trường hợp 4: $x_0 = x_1 = 1 \Rightarrow x = 1$ (Đây là hằng 1)

A	x = f(A)
0	1
1	1

Bảng 16. Bảng thực trị với $x_0 = x_1 = 1$

- Trường hợp 4-dạng chuẩn tuyển: $x = \overline{A}.x_0 + A.x_1 \Rightarrow x = \overline{A}.1 + A.1 = 1$

- Trường hợp 4-dạng chuẩn hội: $x=(A+x_0).(\overline{A}+x_1) \Rightarrow x= (A+1).(\overline{A}+1) = 1$

1.10.3- Hàm 2 biến x = f(B, A)

- Lập bảng thực trị:

B	A	x = f(B, A)
0	0	x_0
0	1	x_1
1	0	x_2
1	1	x_3

Bảng 17. Bảng thực trị với $x_i \in \{0,1\}$

- Công thức đại số dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.x_0 + \overline{B}.A.x_1 + B.\overline{A}.x_2 + B.A.x_3$

- Công thức đại số dạng chuẩn hội: $x = (B+A+x_0).(B+\overline{A}+x_1).(\overline{B}+A+x_2).(\overline{B}+\overline{A}+x_3)$

- Với hàm hai biến, ta liệt kê tất cả trường hợp có thể xảy ra là $2^{2^2} = 2^4 = 16$ trường hợp. Ở đây, chỉ viết công thức dạng chuẩn tuyển.

- Trường hợp 1: $x_0 = x_1 = x_2 = x_3 = 0 \Rightarrow x = 0$ (Đây là hằng 0)

B	A	x = f(B, A)
0	0	0
0	1	0
1	0	0
1	1	0

Bảng 18. Bảng thực trị với $x_0 = x_1 = x_2 = x_3 = 0$

- Trường hợp 1- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.0 + \overline{B}.A.0 + B.\overline{A}.0 + B.A.0 = 0$

- Trường hợp 2: $x_0 = 1$ & $x_1 = x_2 = x_3 = 0 \Rightarrow x = \overline{B}.\overline{A}$ (Đây là cổng NOR)

B	A	x = f(B, A)
0	0	1
0	1	0
1	0	0
1	1	0

Bảng 19. Bảng thực trị với $x_0 = 1$ & $x_1 = x_2 = x_3 = 0$

- Trường hợp 2- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.1 + \overline{B}.A.0 + B.\overline{A}.0 + B.A.0 = \overline{B}.\overline{A}$

- Trường hợp 3: $x_1 = 1$ & $x_0 = x_2 = x_3 = 0$

B	A	x = f(B, A)
0	0	0
0	1	1
1	0	0
1	1	0

Bảng 20. Bảng thực trị với $x_1 = 1$ & $x_0 = x_2 = x_3 = 0$

- Trường hợp 3- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.0 + \overline{B}.A.1 + B.\overline{A}.0 + B.A.0 = \overline{B}.A$

- Trường hợp 4: $x_0 = x_1 = 1$ & $x_2 = x_3 = 0 \Rightarrow x = \overline{B}$

B	A	x = f(B, A)
0	0	1
0	1	1
1	0	0
1	1	0

Bảng 21. Bảng thực trị với $x_0 = x_1 = 1$ & $x_2 = x_3 = 0$

- Trường hợp 4- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.1 + \overline{B}.A.1 + B.\overline{A}.0 + B.A.0 = \overline{B}.\overline{A} + \overline{B}.A = \overline{B}$

- Trường hợp 5: $x_0 = x_1 = x_3 = 0$ & $x_2 = 1$ ⇨ $x = B.\overline{A}$

B	A	x = f(B, A)
0	0	0
0	1	0
1	0	1
1	1	0

Bảng 22. Bảng thực trị với $x_0 = x_1 = x_3 = 0$ & $x_2 = 1$

- Trường hợp 5- dạng chuẩn tuyển: ⇨ $x = \overline{B}.\ \overline{A}.\ 0 + \overline{B}.\ A.\ 0 + B.\ \overline{A}.1 + B.\ A.\ 0 = B.\overline{A}$

- Trường hợp 6: $x_0 = x_2 = 1$ & $x_1 = x_3 = 0$ ⇨ $x = \overline{A}$

B	A	x = f(B, A)
0	0	1
0	1	0
1	0	1
1	1	0

Bảng 23. Bảng thực trị với $x_0 = x_2 = 1$ & $x_1 = x_3 = 0$

- Trường hợp 6- dạng chuẩn tuyển: $x = \overline{B}.\ \overline{A}.\ 1 + \overline{B}.\ A.\ 0 + B.\ \overline{A}.\ 1 + B.\ A.\ 0 = \overline{B}.\overline{A} + B.\overline{A} = \overline{A}$

- Trường hợp 7: $x_1 = x_2 = 1$ & $x_0 = x_3 = 0$ ⇨ $x = B \oplus A$ (Đây là cổng EX-OR)

B	A	x = f(B, A)
0	0	0
0	1	1
1	0	1
1	1	0

Bảng 24. Bảng thực trị với $x_1 = x_2 = 1$ & $x_0 = x_3 = 0$

- Trường hợp 7- dạng chuẩn tuyển: $x = \overline{B}.\ \overline{A}.\ 0 + \overline{B}.\ A.\ 1 + B.\ \overline{A}.\ 1 + B.\ A.\ 0 = \overline{B}.A + B.\overline{A} = B \oplus A$

- Trường hợp 8: $x_0 = x_1 = x_2 = 1$ & $x_3 = 0$ ⇨ $x = \overline{B} + \overline{A}$ (Đây là cổng NAND)

B	A	x = f(B, A)
0	0	1
0	1	1
1	0	1
1	1	0

Bảng 25. Bảng thực trị với $x_0 = x_1 = x_2 = 1$ & $x_3 = 0$

- Trường hợp 8- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.1 + \overline{B}.A.1 + B.\overline{A}.1 + B.A.0 = \overline{B}.\overline{A} + \overline{B}.A + B.\overline{A} = \overline{B}.\overline{A} + \overline{B}.A + B.\overline{A} + \overline{B}.\overline{A} = \overline{B} + \overline{A}$

- Trường hợp 9: $x_3 = 1$ & $x_0 = x_1 = x_2 = 0$ ⇨ $x = B.A$ (Đây là cổng AND là trường hợp đảo của trường hợp 8)

B	A	x = f(B, A)
0	0	0
0	1	0
1	0	0
1	1	1

Bảng 26. Bảng thực trị với $x_3 = 1$ & $x_0 = x_1 = x_2 = 0$

- Trường hợp 9- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.0 + \overline{B}.A.0 + B.\overline{A}.0 + B.A.1 = B.A$

- Trường hợp 10: $x_0 = x_3 = 1$ & $x_1 = x_2 = 0$ ⇨ $x = \overline{A \oplus B}$ (Đây là cổng EX-NOR)

B	A	x = f(B, A)
0	0	1
0	1	0
1	0	0
1	1	1

Bảng 27. Bảng thực trị với $x_0 = x_3 = 1$ & $x_1 = x_2 = 0$

- Trường hợp 10- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.1 + \overline{B}.A.0 + B.\overline{A}.0 + B.A.1 = \overline{B}.\overline{A} + B.A = \overline{A \oplus B}$

- Trường hợp 11: $x_1 = x_3 = 1$ & $x_0 = x_2 = 0$ ⇨ $x = A$

B	A	X = f(B, A)
0	0	0
0	1	1
1	0	0
1	1	1

Bảng 28. Bảng thực trị với $x_1 = x_3 = 1$ & $x_0 = x_2 = 0$

- Trường hợp 11- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.0 + \overline{B}.A.1 + B.\overline{A}.0 + B.A.1 = \overline{B}.A + B.A = A$

- Trường hợp 12: $x_0 = x_1 = x_3 = 1$ & $x_2 = 0$ ⇨ $x = \overline{B} + A$

B	A	X = f(B, A)
0	0	1
0	1	1

1	0	0
1	1	1

Bảng 29. Bảng thực trị với $x_0 = x_1 = x_3 = 1$ & $x_2 = 0$

- Trường hợp 12- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.1 + \overline{B}.A.1 + B.\overline{A}.0 + B.A.1 = \overline{B}.\overline{A} + \overline{B}.A + B.A = \overline{B}.\overline{A} + \overline{B}.A + B.A + \overline{B}.A = \overline{B} + A$

- Trường hợp 13: $x_0 = x_1 = 0$ & $x_2 = x_3 = 1 \Rightarrow x = B$

B	A	x = f(B, A)
0	0	0
0	1	0
1	0	1
1	1	1

Bảng 30. Bảng thực trị với $x_0 = x_1 = 0$ & $x_2 = x_3 = 1$

- Trường hợp 13- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.0 + \overline{B}.A.0 + B.\overline{A}.1 + B.A.1 = B.\overline{A} + B.A = B$

- Trường hợp 14: $x_1 = 0$ & $x_0 = x_2 = x_3 = 1 \Rightarrow x = B + \overline{A}$

B	A	x = f(B, A)
0	0	1
0	1	0
1	0	1
1	1	1

Bảng 31. Bảng thực trị với $x_1 = 0$ & $x_0 = x_2 = x_3 = 1$

- Trường hợp 14- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.1 + \overline{B}.A.0 + B.\overline{A}.1 + B.A.1 = \overline{B}.\overline{A} + B.\overline{A} + B.A = \overline{B}.\overline{A} + B.\overline{A} + B.A + B.\overline{A} = B + \overline{A}$

- Trường hợp 15: $x_1 = x_2 = x_3 = 1$ & $x_0 = 0 \Rightarrow x = A+B$ (Đây chính là cổng OR – là đảo của trường hợp 2, cổng NOR)

B	A	x = f(B, A)
0	0	0
0	1	1
1	0	1
1	1	1

Bảng 32. Bảng thực trị với $x_1 = x_2 = x_3 = 1$ & $x_0 = 0$

- Trường hợp 15- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.0 + \overline{B}.A.1 + B.\overline{A}.1 + B.A.1 = \overline{B}.A + B.\overline{A} + B.A = \overline{B}.A + B.A + B.\overline{A} + B.A = A+B$

- Trường hợp 16: $x_0 = x_1 = x_2 = x_3 = 1 \Rightarrow x = 1$ (Đây là hằng 1)

B	A	x = f(B, A)
0	0	1
0	1	1
1	0	1
1	1	1

Bảng 33. Bảng thực trị với $x_0 = x_1 = x_2 = x_3 = 1$

- Trường hợp 16- dạng chuẩn tuyển: $x = \overline{B}.\overline{A}.1 + \overline{B}.A.1 + B.\overline{A}.1 + B.A.1 = \overline{B}.\overline{A} + \overline{B}.A + B.\overline{A} + B.A = \overline{B} + B = 1$

1.10.4- Hàm 3 biến x = f(C, B, A)

- Lập bảng thực trị:

C	B	A	X = f(C, B, A)
0	0	0	x_0
0	0	1	x_1
0	1	0	x_2
0	1	1	x_3
1	0	0	x_4
1	0	1	x_5
1	1	0	x_6
1	1	1	x_7

Bảng 34. Bảng thực trị với $x_i \in \{0,1\}$

- Công thức đại số dạng chuẩn tuyển: $x = \overline{C}.\overline{B}.\overline{A}.x_0 + \overline{C}.\overline{B}.A.x_1 + \overline{C}.B.\overline{A}.x_2 + \overline{C}.B.A.x_3 + C.\overline{B}.\overline{A}.x_4 + C.\overline{B}.A.x_5 + C.B.\overline{A}.x_6 + C.B.A.x_7$

- Công thức đại số dạng chuẩn hội: $x = (C+B+A+x_0).(C+B+\overline{A}+x_1).(C+\overline{B}+A+x_2).(C+\overline{B}+\overline{A}+x_3).(\overline{C}+B+A+x_4).(\overline{C}+B+\overline{A}+x_5).(\overline{C}+\overline{B}+A+x_6).(\overline{C}+\overline{B}+\overline{A}+x_7)$

- Với hàm ba biến, tất cả trường hợp có thể xảy ra là $2^{2^3} = 2^8 = 256$ trường hợp.

1.10.5- Hàm 4 biến x = f(D, C, B, A)

- Lập bảng thực trị:

D	C	B	A	x=f(D, C, B,A)
0	0	0	0	x_0
0	0	0	1	x_1
0	0	1	0	x_2

0	0	1	1	x_3
0	1	0	0	x_4
0	1	0	1	x_5
0	1	1	0	x_6
0	1	1	1	x_7
1	0	0	0	x_8
1	0	0	1	x_9
1	0	1	0	x_{10}
1	0	1	1	x_{11}
1	1	0	0	x_{12}
1	1	0	1	x_{13}
1	1	1	0	x_{14}
1	1	1	1	x_{15}

Bảng 35. Bảng thực trị với $x_i \in \{0,1\}$

- Công thức đại số dạng chuẩn tuyển: $x = \overline{D}.\overline{C}.\overline{B}.\overline{A}.x_0 + \overline{D}.\overline{C}.\overline{B}.A.x_1 + \overline{D}.\overline{C}.B.\overline{A}.x_2 + \overline{D}.\overline{C}.B.A.x_3 + \overline{D}.C.\overline{B}.\overline{A}.x_4 + \overline{D}.C.\overline{B}.A.x_5 + \overline{D}.C.B.\overline{A}.x_6 + \overline{D}.C.B.A.x_7 + D.\overline{C}.\overline{B}.\overline{A}.x_8 + D.\overline{C}.\overline{B}.A.x_9 + D.\overline{C}.B.\overline{A}.x_{10} + D.\overline{C}.B.A.x_{11} + D.C.\overline{B}.\overline{A}.x_{12} + D.C.\overline{B}.A.x_{13} + D.C.B.\overline{A}.x_{14} + D.C.B.A.x_{15}$

- Công thức đại số dạng chuẩn hội: $x = (D+C+B+A+x_0).(D+C+B+\overline{A}+x_1).(D+C+\overline{B}+A+x_2).(D+C+\overline{B}+\overline{A}+x_3).(D+\overline{C}+B+A+x_4).(D+\overline{C}+B+\overline{A}+x_5).(D+\overline{C}+\overline{B}+A+x_6).(D+\overline{C}+\overline{B}+\overline{A}+x_7).(\overline{D}+C+B+A+x_8).(\overline{D}+C+B+\overline{A}+x_9).(\overline{D}+C+\overline{B}+A+x_{10}).(\overline{D}+C+\overline{B}+\overline{A}+x_{11}).(\overline{D}+\overline{C}+B+A+x_{12}).(\overline{D}+\overline{C}+B+\overline{A}+x_{13}).(\overline{D}+\overline{C}+\overline{B}+A+x_{14}).(\overline{D}+\overline{C}+\overline{B}+\overline{A}+x_{15})$

- Với hàm bốn biến, tất cả trường hợp có thể xảy ra là $2^{2^4} = 2^{16} = 65536$ trường hợp.

1.10.6- Hàm N biến x = f(A_{n-1}, A_{n-2}, …, A_1, A_0)

- Lập bảng thực trị:

A_{n-1}	A_{n-2}	…	A_0	$x=f(A_{n-1}, A_{n-2}, \ldots, A_0)$
0	0	…	0	X_0
0	0	…	1	x_1
. . .	. . .	. . .	. . .	. . .
1	1	…	0	x_{n-2}
1	1	…	1	x_{n-1}

Bảng 36. Bảng thực trị với $x_i \in \{0,1\}$

- Công thức đại số dạng chuẩn tuyển: $x = \overline{A}_{n-1}.\overline{A}_{n-2}\ldots\overline{A}_1.\overline{A}_0.x_0 + \overline{A}_{n-1}.\overline{A}_{n-2}\ldots\overline{A}_1.\overline{A}_0.x_1 + \overline{A}_{n-1}.\overline{A}_{n-2}\ldots A_1.\overline{A}_0.x_2 + \ldots + A_{n-1}.A_{n-2}\ldots A_1.A_0.x_{n-1}$.

- Công thức đại số dạng chuẩn hội: $x = (A_{n-1}+A_{n-2}+\ldots+A_1+A_0+x_0)\ .\ (A_{n-1}+A_{n-2}+\ldots+A_1+\overline{A}_0+x_1).(A_{n-1}+A_{n-2}+\ldots+\overline{A}_1+A_0+x_2)\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (\overline{A}_{n-1}+\overline{A}_{n-2}+..\overline{A}_1+\overline{A}_0+x_{n-1})$

- Với hàm N biến, tất cả trường hợp có thể xảy ra là 2^{2^N} trường hợp.

- Nhận xét 1: từ dạng chuẩn tuyển, ta thu được dạng chuẩn hội bằng cách lấy phủ định hai lần của dạng chuẩn tuyển kết hợp với định lý DeMorgan.

- Nhận xét 2: Trong dạng chuẩn tuyển, do phép toán AND với 0 thì sẽ bằng 0, và AND với 1 thì bằng chính nó, nên ta chỉ viết những biểu thức logic tương ứng với những giá trị x_i=1, còn những biểu thức logic ứng với x_i=0 thì không viết vào công thức.

- Nhận xét 3: Trong dạng chuẩn hội, do phép toán OR với 0 thì sẽ bằng chính nó, và OR với 1 thì bằng 1, nên ta chỉ viết những biểu thức logic tương ứng với những giá trị x_i=0, còn những biểu thức logic ứng với x_i=1 thì không viết vào công thức.

- Ví dụ minh họa:

D	C	B	A	x	Chuẩn tuyển	Chuẩn hội
0	0	0	0	1	$\overline{D}.\overline{C}.\overline{B}.\overline{A}$	
0	0	0	1	1	$\overline{D}.\overline{C}.\overline{B}.A$	
0	0	1	0	1	$\overline{D}.\overline{C}.B.\overline{A}$	
0	0	1	1	0		$D+C+\overline{B}+\overline{A}$
0	1	0	0	1	$\overline{D}.C.\overline{B}.\overline{A}$	
0	1	0	1	1	$\overline{D}.C.\overline{B}.A$	
0	1	1	0	1	$\overline{D}.C.B.\overline{A}$	
0	1	1	1	0		$D+\overline{C}+\overline{B}+\overline{A}$
1	0	0	0	1	$D.\overline{C}.\overline{B}.\overline{A}$	

1	0	0	1	1	$D.\overline{C}.\overline{B}.A$	
1	0	1	0	1	$D.\overline{C}.B.\overline{A}$	
1	0	1	1	0		$\overline{D}+C+\overline{B}+\overline{A}$
1	1	0	0	0		$\overline{D}+\overline{C}+B+A$
1	1	0	1	0		$\overline{D}+\overline{C}+B+\overline{A}$
1	1	1	0	0		$\overline{D}+\overline{C}+\overline{B}+A$
1	1	1	1	0		$\overline{D}+\overline{C}+\overline{B}+\overline{A}$

Bảng 37. Cách viết công thức đại số dạng chuẩn tuyển và chuẩn hội

- Ta viết x dưới dạng chuẩn tuyển như sau:
  $x = \overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.A + \overline{D}.\overline{C}.B.\overline{A} + \overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.\overline{B}.A + \overline{D}.C.B.\overline{A} + D.\overline{C}.\overline{B}.\overline{A} + D.\overline{C}.\overline{B}.A + D.\overline{C}.B.\overline{A}$

- Ta viết x dưới dạng chuẩn hội như sau:
  $x = (D+C+\overline{B}+\overline{A}).(D+\overline{C}+\overline{B}+\overline{A}).(\overline{D}+C+\overline{B}+\overline{A}).(\overline{D}+\overline{C}+B+A).(\overline{D}+\overline{C}+\overline{B}+A).(\overline{D}+\overline{C}+\overline{B}+\overline{A})$

- Trong những trường hợp biểu thức đại số không ở dạng chuẩn, thì ta phải dùng các định lý Bool và định lý DeMorgan để đưa về dạng chuẩn. Những trường hợp không dùng dạng chuẩn như chỉ dùng một loại cổng NAND hay một loại cổng NOR, hay dùng hai loại cổng EX-OR và EX-NOR chỉ sử dụng với những yêu cầu cụ thể trong những trường hợp đặc biệt.

- Những công thức viết dưới hai dạng chuẩn trên là những công thức viết đầy đủ, và người thiết kế mạch số phải có nhiệm vụ rút gọn mạch để giảm số cổng hay số vi mạch sử dụng cho mạch đó ở mức tối thiểu.

1.11- Rút Gọn Một Biểu Thức Bằng Phương Pháp Đại Số

- Thường phương pháp này được áp dụng cho dạng chuẩn tuyển.

1.11.1.1- Rút gọn từ dạng đầy đủ, hay từ bảng thực trị viết ra

- Lấy một ví dụ từ bảng thực trị trên, ta có:

  $x = \overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.A + \overline{D}.\overline{C}.B.\overline{A} + \overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.\overline{B}.A + \overline{D}.C.B.\overline{A} + D.\overline{C}.\overline{B}.\overline{A} + D.\overline{C}.\overline{B}.A + D.\overline{C}.B.\overline{A}$

  Nhận xét: lấy $\overline{D}.\overline{C}.\overline{B}.\overline{A}$ ra xét, xem có bao nhiêu biểu thức khác nó một bit, ta sẽ thấy đó là các biểu thức $\overline{D}.\overline{C}.\overline{B}.A$, $\overline{D}.C.\overline{B}.\overline{A}$, $\overline{D}.C.\overline{B}.A$

  ta có thể rút gọn 4 biểu thức trên như sau

  $*\overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.A + \overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.\overline{B}.A = \overline{D}.\overline{C}.\overline{B}.(\overline{A} + A) + \overline{D}.C.\overline{B}.(\overline{A} + A) = \overline{D}.\overline{C}.\overline{B} + \overline{D}.C.\overline{B} = \overline{D}.\overline{B}.(\overline{C} + C) = \overline{D}.\overline{B}$

  hay viết một cách ngắn gọn hơn là

  $*\overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.A + \overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.\overline{B}.A = \overline{D}.\overline{B}.(\overline{C}.\overline{A} + \overline{C}.A + C.\overline{A} + C.A) = \overline{D}.\overline{B}$

  tương tự ta sẽ có các cặp

  $*\overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.B.\overline{A} + \overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.B.\overline{A} = \overline{D}.\overline{A}.(\overline{C}.\overline{B} + \overline{C}.B + C.\overline{B} + C.B) = \overline{D}.\overline{A}$

  $*\overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.A + D.\overline{C}.\overline{B}.\overline{A} + D.\overline{C}.\overline{B}.A = \overline{C}.\overline{B}.(\overline{D}.\overline{A} + \overline{D}.A + D.\overline{A} + D.A) = \overline{C}.\overline{B}$

  $*\overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.B.\overline{A} + D.\overline{C}.\overline{B}.\overline{A} + D.\overline{C}.B.\overline{A} = \overline{C}.\overline{A}.(\overline{D}.\overline{B} + \overline{C}.B + C.\overline{B} + C.B) = \overline{C}.\overline{A}$

  như vậy, ta đã dùng $\overline{D}.\overline{C}.\overline{B}.\overline{A}$ 4 lượt, $\overline{D}.\overline{C}.\overline{B}.A$ hai lượt, $\overline{D}.\overline{C}.B.\overline{A}$ hai lượt, $\overline{D}.C.\overline{B}.\overline{A}$ hai lượt, $D.\overline{C}.\overline{B}.\overline{A}$ ba lượt ⇨ rút gọn x như sau

  $x = \overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.A + \overline{D}.\overline{C}.B.\overline{A} + \overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.\overline{B}.A + \overline{D}.C.B.\overline{A} + D.\overline{C}.\overline{B}.\overline{A} + D.\overline{C}.\overline{B}.A + D.\overline{C}.B.\overline{A} = (\overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.\overline{A} + \overline{D}.\overline{C}.\overline{B}.\overline{A}) + (\overline{D}.\overline{C}.\overline{B}.A + \overline{D}.\overline{C}.\overline{B}.A) + (\overline{D}.\overline{C}.B.\overline{A} + \overline{D}.\overline{C}.B.\overline{A}) + (\overline{D}.C.\overline{B}.\overline{A} + \overline{D}.C.\overline{B}.\overline{A}) + \overline{D}.C.\overline{B}.A + \overline{D}.C.B.\overline{A} + (D.\overline{C}.\overline{B}.\overline{A} + D.\overline{C}.\overline{B}.\overline{A} + D.\overline{C}.\overline{B}.\overline{A}) + D.\overline{C}.\overline{B}.A + D.\overline{C}.B.\overline{A} = \overline{D}.\overline{B} + \overline{D}.\overline{A} + \overline{C}.\overline{B} + \overline{C}.\overline{A}$

- Lấy một ví dụ khác

  $x = D.C.\overline{B}.\overline{A} + D.C.\overline{B}.A + \overline{D}.C.\overline{B}.A + D.\overline{C}.\overline{B}.A + \overline{D}.\overline{C}.B.A + \overline{D}.\overline{C}.B.\overline{A}$

  Nhận xét: lấy $D.C.\overline{B}.\overline{A}$ ra xét, xem có bao nhiêu biểu thức khác nó một bit, ta thấy chỉ có $D.C.\overline{B}.A$ ⇨ ta có thể rút gọn biểu thức trên thành

  $*D.C.\overline{B}.\overline{A} + D.C.\overline{B}.A = D.C.\overline{B}(\overline{A} + A) = D.C.\overline{B}$

  tương tự ta có các cặp:

* $\overline{D}.C.\overline{B}.A + D.C.\overline{B}.A + D.\overline{C}.\overline{B}.A = \overline{D}.C.\overline{B}.A + (D.C.\overline{B}.A + D.C.\overline{B}.A) + D.\overline{C}.\overline{B}.A = C.\overline{B}.A.(\overline{D}+D) + D.\overline{B}.A.(C+\overline{C}) = C.\overline{B}.A + D.\overline{B}.A$

* $\overline{D}.\overline{C}.B.A + \overline{D}.\overline{C}.B.\overline{A} = \overline{D}.\overline{C}.B.(A+\overline{A}) = \overline{D}.\overline{C}.B$

Như vậy, ta đã dùng $D.C.\overline{B}.A$ ba lượt, ⇨ ta rút gọn x như sau:

$x = D.C.\overline{B}.\overline{A} + D.C.\overline{B}.A + \overline{D}.C.\overline{B}.A + D.\overline{C}.\overline{B}.A + \overline{D}.\overline{C}.B.A + \overline{D}.\overline{C}.B.\overline{A} = D.C.\overline{B}.\overline{A} + (D.C.\overline{B}.A + D.C.\overline{B}.A + D.C.\overline{B}.A) + \overline{D}.C.\overline{B}.A + D.\overline{C}.\overline{B}.A + \overline{D}.\overline{C}.B.A + \overline{D}.\overline{C}.B.\overline{A} = D.C.\overline{B} + C.\overline{B}.A + D.\overline{B}.A + \overline{D}.\overline{C}.B$

1.11.1.2- Rút gọn từ dạng không đầy đủ

- Lấy một ví dụ:

  $x = C.\overline{B} + \overline{D}.C.\overline{B} + \overline{D}.C.A + \overline{D}.C.B.A + C.\overline{B}.A + D.\overline{B}.A + D.\overline{C}.\overline{B}.A$

  Nhận xét: lấy biểu thức đã rút gọn nhất là $C.\overline{B}$ và tìm trong đó có biểu thức nào có chứa $C.\overline{B}$, ta có $\overline{D}.C.\overline{B}$ và $C.\overline{B}.A$, ⇨ $C.\overline{B} + \overline{D}.C.\overline{B} + C.\overline{B}.A = C.\overline{B}.(1 + \overline{D} + A) = C.\overline{B}$

  lấy biểu thức rút gọn còn lại, mà chưa được liệt kê ở phần trên là $\overline{D}.\overline{C}.A$ và tiếp tục tìm xem có biểu thức nào chứa $\overline{D}.\mathbf{C}.A$, ta có $\overline{D}.\mathbf{C}.B.A$ ⇨ $\overline{D}.\mathbf{C}.A + \overline{D}.\mathbf{C}.B.A = \overline{D}.\mathbf{C}.A.(1 + \mathbf{B}) = \overline{D}.\mathbf{C}.A$

  tương tự ta có cặp $D.\overline{B}.A$ và $D.\overline{C}.\overline{B}.A$ ⇨ $D.\overline{B}.A + D.\overline{C}.\overline{B}.A = D.\overline{B}.A.(1+\overline{C}) = D.\overline{B}.A$

  Vậy $x = C.\overline{B} + \overline{D}.C.\overline{B} + \overline{D}.C.A + \overline{D}.C.B.A + C.\overline{B}.A + D.\overline{B}.A + D.\overline{C}.\overline{B}.A = C.\overline{B}.(1 + \overline{D} + A) + \overline{D}.C.A.(1 + B) + D.\overline{B}.A.(1+\overline{C}) = C.\overline{B} + \overline{D}.C.A + D.\overline{B}.A$

1.11.1.3- Rút gọn từ dạng chuẩn hội

- Ta nhân các biểu thức lại với nhau. xong áp dụng các phương pháp nêu trên

- Lấy một ví dụ: $x = (A + C).(B + C + \overline{D}).D = A.B.D + A.C.D + A.\overline{D}.D + B.C.D + C.C.D + C.\overline{D}.D = A.B.D + A.C.D + B.C.D + C.D$

  Áp dụng phương pháp trên ta có $x = C.D.(1 + A + B) + A.B.D = C.D + A.B.D$

1.11.1.4- Rút gọn từ dạng chưa được chuẩn hoá

- Áp dụng định lý DeMorgan và các định lý đã học để đưa về dạng chuẩn rồi rút gọn

- Lấy một ví dụ: $x = \overline{A}.C.(\overline{\overline{A}.B.D}) + \overline{A}.B.\overline{C}.\overline{D} + A.\overline{B}.C = \overline{A}.C.(A + \overline{B} + \overline{D}) + \overline{A}.B.\overline{C}.\overline{D} + A.\overline{B}.C = \overline{A}.C.A + \overline{A}.C.\overline{B} + \overline{A}.C.\overline{D} + \overline{A}.B.\overline{C}.\overline{D} + A.\overline{B}.C = \overline{A}.C.\overline{B} + \overline{A}.C.\overline{D} + \overline{A}.B.\overline{C}.\overline{D} + A.\overline{B}.C = A.\overline{B}.C.(1 + \overline{D}) + \overline{B}.C.(\overline{A} + A) + \overline{A}.C.\overline{D} = \overline{B}.C + A.\overline{B}.C + \overline{A}.C.\overline{D}$

1.12- Rút Gọn Một Biểu Thức Bằng Phương Pháp Biểu Đồ Karnaugh – Dạng Chuẩn Tuyển

- Trên cơ sở của các định lý Bool, nhà toán Học Karnaugh đã đưa ra một phương pháp trực quan để rút gọn một biểu thức Bool, phương pháp có tên là biểu đồ Karnaugh.

- Phương pháp này chỉ áp dụng được cho các hàm 6 biến trở xuống, và dễ áp dụng nhất là 4 biến trở xuống. Trong tài liệu này, ta sẽ xét từ trường hợp 1 biến đến 4 biến, trước khi đi vào phương pháp rút gọn tổng quát.

- Công thức đại số trong các phần dưới được viết dưới dạng chuẩn tuyển, còn dạng chuẩn hội sẽ đề cập ở phần sau.

1.12.1- Biểu đồ Karnaugh cho hàm 1 biến x = f(A)

Hàm 1 biến chỉ có 4 trường hợp và quá đơn giản, nên không dùng biểu đồ Karnaugh để đơn giản.

1.12.2- Biểu đồ Karnaugh cho hàm 2 biến x = f(B, A)

- Biểu đồ Karnaugh một hình vuông có 4 ô như sau:

$\overline{B}.\overline{A}.x_0$	$\overline{B}.A.x_1$
$B.\overline{A}.x_2$	$B.A.x_3$

Bảng 38. Biểu đồ Karnaugh cho hàm hai biến

- Viết như vậy thì rườm rà và khó rút gọn, nên đưa các giá trị $\overline{B}$, B, $\overline{A}$, A ra ngoài như hình dưới:

	$\overline{A}$	A
$\overline{B}$	x_0	x_1
B	x_2	x_3

Bảng 39. Cách biểu diễn khác của biểu đồ Karnaugh

- ➢ Cũng có một cách biểu diễn khác, thay cho $\overline{A}$, $\overline{B}$ là bit 0 và thay cho A, B là bit 1.

B \ A	0	1
0	x_0	x_1
1	x_2	x_3

Bảng 40. Cách biểu diễn khác của biểu đồ Karnaugh

- ➢ Hai dạng biểu diễn sau thường được sử dụng hơn.
- ➢ Khi áp dụng vào một trường hợp cụ thể, thì ta thay các giá trị $x_i = 0$ hay $x_i = 1$ vào các ô trong hình vuông.

Một số nguyên tắc khi xây dựng biểu đồ Karnaugh hai biến:

- ➢ Biểu đồ Karnaugh cho hàm hai biến là 1 hình vuông gồm 4 ô.
- ➢ Mỗi ô ứng với 1 dòng trong bảng thực trị (tích của 2 biến B, A và x_i).
- ➢ Mỗi ô nằm kế cận 2 ô.
- ➢ Hai ô cạnh nhau chỉ khác nhau duy nhất 1 bit.
- ➢ Ô nằm riêng lẻ một mình có giá trị $x_i = 1$, khoanh chúng lại thành một ***vòng*** gồm 1 ô, được gọi là LOOP1, không rút gọn được biến nào.
- ➢ Hai ô nằm cạnh nhau có giá trị là $x_i = 1$, khoanh chúng lại thành một ***vòng*** gồm 2 ô, được gọi là LOOP2, thì rút gọn được một biến.

- Bốn ô nằm cạnh nhau có giá trị là $x_i = 1$, khoanh chúng lại thành một ***vòng*** gồm 4 ô, được gọi là LOOP4, thì rút gọn được hai biến.

Xét một số trường hợp của hàm hai biến sau:

	$\overline{A}$	A
$\overline{B}$	1	1
B	1	1

Bảng 41. Một biểu đồ Karnaugh với LOOP4

- Hàm 2 biến có một LOOP4 bị rút gọn hết 2 biến ⇨ x = 1.

	$\overline{A}$	A
$\overline{B}$	0	1
B	1	1

Bảng 42. Một biểu đồ Karnaugh với hai LOOP2

- Như trường hợp này có hai LOOP2, LOOP2 thứ nhất gồm có $A.\overline{B} + A.B$, thì bỏ biến B chỉ còn lại biến A. LOOP2 thứ hai gồm có $\overline{A}.B + A.B$, thì bỏ biến A chỉ còn lại biến B, như vậy công thức rút gọn được là x = A + B. Hay dễ nhìn hơn là LOOP2 thứ nhất là cột A và LOOP2 thứ 2 là cột B ⇨ x = A + B.

- Từ biểu đồ Karnaugh, ta có thể áp dụng vào việc rút gọn bằng phương pháp đại số. Lấy ví dụ trên, biểu thức chưa rút gọn là $x = B.A + B.\overline{A} + \overline{B}.A$, căn cứ vào biểu đồ Karnaugh, thì ô B.A được dùng hai lần, ta áp dụng như sau: $x = B.A + B.\overline{A} + \overline{B}.A = B.A + B.A + B.\overline{A} + \overline{B}.A = B.(A+\overline{A}) + A.(B+\overline{B}) = B+A$.

	$\overline{A}$	A
$\overline{B}$	0	$\boxed{1}$
B	$\boxed{1}$	0

Bảng 43. Một biểu đồ Karnaugh với hai LOOP1

- Như trường hợp này có hai LOOP1, LOOP1 thứ nhất là $A.\overline{B}$ và LOOP1 thứ 2 là $\overline{A}.B$, trường hợp này không thể rút gọn
  ⇨ $x = A.\overline{B} + \overline{A}.B$

1.12.3- Biểu đồ Karnaugh cho hàm 3 biến x = f(C, B, A)

- Biểu đồ Karnaugh một hình chữ nhật có 8 ô như sau:

$\overline{C}.\overline{B}.\overline{A}.x_0$	$\overline{C}.\overline{B}.A.x_1$	$\overline{C}.B.A.x_3$	$\overline{C}.B.\overline{A}.x_2$
$C.\overline{B}.\overline{A}.x_4$	$C.\overline{B}.A.x_5$	$C.B.A.x_7$	$C.B.\overline{A}.x_6$

Hình 68. Biểu đồ Karnaugh cho hàm ba biến

- Viết như vậy thì rườm rà và khó rút gọn, nên đưa các giá trị $\overline{B}$, B, $\overline{A}$, A ra ngoài như hình dưới:

	$\overline{B}.\overline{A}$	$\overline{B}.A$	$B.A$	$B.\overline{A}$
$\overline{C}$	x_0	x_1	x_3	x_2
C	x_4	x_5	x_7	x_6

Bảng 44. Cách biểu diễn khác của biểu đồ Karnaugh

- Cũng có một cách biểu diễn khác, thay cho $\overline{A}$, $\overline{B}$ là bit 0 và thay cho A, B là bit 1.

B.A / C	00	01	11	10
0	x_0	x_1	x_3	x_2
1	x_4	x_5	x_7	x_6

Bảng 45. Cách biểu diễn khác của biểu đồ Karnaugh

- Hai dạng biểu diễn sau thường được sử dụng hơn.

- Khi áp dụng vào một trường hợp cụ thể, thì ta thay các giá trị x_i = 0 hay x_i = 1 vào các ô trong hình chữ nhật.

Một số nguyên tắc khi xây dựng biểu đồ Karnaugh ba biến:

- Biểu đồ Karnaugh cho hàm hai biến là 1 hình chữ nhật gồm 8 ô.

- Mỗi ô ứng với 1 dòng trong bảng thực trị (tích của 3 biến C, B, A và x_i).

- Mỗi ô nằm kế cận 3 ô.

- Hai ô cạnh nhau chỉ khác nhau duy nhất 1 bit.

- Ô nằm riêng lẻ một mình có giá trị x_i = 1, khoanh chúng lại thành một ***vòng*** gồm 1 ô, được gọi là LOOP1, không rút gọn được biến nào.

- Hai ô nằm cạnh nhau có giá trị là x_i = 1, khoanh chúng lại thành một ***vòng*** gồm 2 ô, được gọi là LOOP2, thì rút gọn được một biến.

- Bốn ô nằm cạnh nhau có giá trị là x_i = 1, khoanh chúng lại thành một ***vòng*** gồm 4 ô, được gọi là LOOP4, thì rút gọn được hai biến.

- Tám ô nằm cạnh nhau có giá trị là x_i = 1, khoanh chúng lại thành một ***vòng*** gồm 8 ô, được gọi là LOOP8, thì rút gọn được ba biến.

Xét một trường hợp của hàm ba biến sau:

- Cho một hàm ba biến với bảng thực trị sau:

C	B	A	x = f(C, B, A)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Bảng 46. Bảng thực trị của 1 hàm ba biến

- Từ bảng thực trị, ta viết công thức đại số: $x = \overline{C}.\overline{B}.\overline{A}.0 + \overline{C}.\overline{B}.A.1 + \overline{C}.B.\overline{A}.1 + \overline{C}.B.A.1 + C.\overline{B}.\overline{A}.0 + C.\overline{B}.A.1 + C.B.\overline{A}.0 + C.B.A.1 \Rightarrow x = \overline{C}.\overline{B}.A + \overline{C}.B.\overline{A} + \overline{C}.B.A + C.\overline{B}.A + C.B.A$

- Ta có biểu đồ Karnaugh như hình dưới

	$\overline{B}.\overline{A}$	$\overline{B}.A$	$B.A$	$B.\overline{A}$
$\overline{C}$	0	1	1	1
C	0	1	1	0

Bảng 47. Một biểu đồ Karnaugh gồm 1 LOOP4 và 1 LOOP2

- ➢ Hay biểu diễn dưới hình thức khác

C \ BA	00	01	11	10
0	0	1	1	1
1	0	1	1	0

Bảng 48. Cách biểu diễn khác của biểu đồ Karnaugh

- ➢ Với biểu đồ Karnaugh gồm 1 LOOP4 và 1 LOOP2 ⇨ $x = A + \overline{C}.B$.

1.12.4- Biểu đồ Karnaugh cho hàm 4 biến x = f(D, C, B, A)

- Để tổng quát cách xây dựng biểu đồ Karnaugh, ta đưa ra một số nguyên tắc sau:

  - ➢ Biểu đồ Karnaugh cho hàm 4 biến là 1 hình vuông có 16 ô.
  - ➢ Mỗi ô ứng với 1 dòng trong bảng thực trị (tích của 4 biến D, C, B, A và x_i).
  - ➢ Mỗi ô nằm kế cận 4 ô, như ô x_5 kề 4 ô x_1, x_4, x_7, x_{13}, ô x_6 kề với 4 ô x_2, x_4, x_7, x_{14}, ô x_0 kề 4 ô x_1, x_4, x_8, x_{10}, ...
  - ➢ Hai ô cạnh nhau chỉ khác nhau duy nhất 1 bit.
  - ➢ LOOP là tập hợp các ô có giá trị x_i = 1 nằm cạnh nhau theo nguyên tắc 2^n, với n = 0, 1, 2, 3, 4 ⇨ LOOP1, LOOP2, LOOP4, LOOP8, LOOP16.
  - ➢ Ô nằm riêng lẻ một mình có giá trị x_i = 1, khoanh chúng lại thành một ***vòng*** gồm 1 ô, được gọi là LOOP1, không rút gọn được biến nào.

- ➢ Hai ô nằm cạnh nhau có giá trị là $x_i = 1$, khoanh chúng lại thành một ***vòng*** gồm 2 ô**,** được gọi là LOOP2, thì rút gọn được một biến.

- ➢ Bốn ô nằm cạnh nhau có giá trị là $x_i = 1$, khoanh chúng lại thành một ***vòng*** gồm 4 ô, được gọi là LOOP4, thì rút gọn được hai biến. Trong biểu đồ Karnaugh 4 biến, LOOP4 là 4 ô tạo thành hình vuông, hay tạo thành một cột, một hàng hay 4 ô ở 4 góc.

- ➢ Tám ô nằm cạnh nhau có giá trị là $x_i = 1$, khoanh chúng lại thành một ***vòng*** gồm 8 ô, được gọi là LOOP8, thì rút gọn được ba biến. Trong biểu đồ Karnaugh 4 biến, LOOP8 là 8 ô tạo thành hai hàng hay hay cột.

- ➢ Mười sáu ô nằm cạnh nhau có giá trị là $x_i = 1$, khoanh chúng lại thành một ***vòng*** gồm 16 ô, được gọi là LOOP16, thì rút gọn được bốn biến.

Từ bảng thực trị, hay từ công thức đại số, ta xây dựng biểu đồ Karnaugh theo các nguyên tắc trên, tuỳ thuộc vào thứ tự D, C, B, A chúng ta sẽ có các biểu đồ Karnaugh khác nhau về mặt hình thức, nhưng kết quả vẫn như nhau. Như một số biểu đồ Karnaugh dưới đây.

DC \ BA	00	10	11	01
00	x_0	x_2	x_3	x_1
10	x_8	x_{10}	x_{11}	x_9
11	x_{12}	x_{14}	x_{15}	x_{13}
01	x_4	x_6	x_7	x_5

hay

DC \ BA	11	10	00	01
01	x_7	x_6	x_4	x_5
11	x_{15}	x_{14}	x_{12}	x_{13}
10	x_{11}	x_{10}	x_8	x_9
00	x_3	x_2	x_0	x_1

hay ...

Bảng 49. Một số cách biểu diễn khác nhau của khác nhau của biểu đồ Karnaugh

- Trong tài liệu này, để thống nhất về mặt ký hiệu, biểu đồ Karnaugh được xây dựng một trong hai dạng với thứ tự sau:

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5	x_7	x_6
11	x_{12}	x_{13}	x_{15}	x_{14}
10	x_8	x_9	x_{11}	x_{10}

Hay

	$\overline{B}.\overline{A}$	$\overline{B}.A$	$B.A$	$B.\overline{A}$
$\overline{D}.\overline{C}$	x_0	x_1	x_3	x_2
$\overline{D}.C$	x_4	x_5	x_7	x_6
$D.C$	x_{12}	x_{13}	x_{15}	x_{14}
$D.\overline{C}$	x_8	x_9	x_{11}	x_{10}

Bảng 50. Cách biểu diễn biểu đồ Karnaugh dùng trong tài liệu

- Và để cho gọn, những ô có giá trị $x_i = 0$ sẽ để trống, chỉ đưa vào những ô có giá trị $x_i = 1$.

- Hàng số 1 là $\overline{D}.\overline{C}$, hàng số 2 là $\overline{D}.C$, hàng số 3 là $D.C$, hàng số 4 là $D.\overline{C}$.

- Cột số 1 là $\overline{B}.\overline{A}$, cột số 2 là $\overline{B}.A$, cột số 3 là $B.A$, cột số 4 là $B.\overline{A}$.

- Giải thuật: (bài toán tô màu, hay bài toán tìm phủ tối tiểu) - bao gồm 4 bước)

  - Xét bài toán tô màu được phát biểu như sau: cho 1 hình gồm nhiều ô, những ô cùng nhóm được tô 1 màu, một ô có thể thuộc nhiều nhóm, tô làm sao kín hình ban đầu với số màu ít nhất và lượng màu dùng cho mỗi màu là ít nhất.

  - Giải thuật rút gọn trong biểu đồ Karnaugh thực chất là bài toán tô màu (hay là bài toán tìm phủ tối tiểu), với mỗi LOOP là 1 màu, số ô trong mỗi LOOP là khối lượng của mỗi màu. Rút gọn làm sao số LOOP ít nhất, và mỗi LOOP thì có ít biến nhất.

- Bước 1: liệt kê tất cả các LOOP8, LOOP4, LOOP2, LOOP1, lưu ý là không được có trường hợp LOOPi $\subseteq$ LOOPj, và viết công thức rút gọn cho mỗi LOOP.
- Bước 2: tìm ô nào chỉ thuộc duy nhất 1 LOOP, ta chọn LOOP đó, lặp lại bước này đến khi nào khi nào không còn ô nào có tính chất trên. Nếu thực hiện xong bước này mà tô kín biểu đồ Karnaugh ban đầu, thì ta có duy nhất 1 kết quả rút gọn , và kết quả này là tối giản.
- Bước 3:

  - Nếu sau bước 2 mà biểu đồ Karnaugh vẫn chưa được tô kín, chọn tuỳ ý 1 LOOP , lặp lại bước này cho đến khi tô kín biểu đồ Karnaugh ban đầu. Ta có một kết quả rút gọn, nhưng do những ô ở bước này thuộc ít nhất là 2 LOOP trở lên, nên việc lựa chọn tuỳ ý 1 LOOP chứa những ô thuộc loại này để tô, dẫn đến kết quả là chưa bảo đảm tối giản.

  - Do đó, ta phải xét đến phản đề, là để tô ô được chọn tuỳ ý mà không dùng LOOP ở bước (3.i), thì phải dùng đến LOOP kia và ta sẽ có một kết quả khác. Và việc xét phản đề sẽ đi theo thứ tự ngược với bước (3.i).
- Bước 4:

  - Sau bước 3, ta sẽ có nhiều kết quả rút gọn khác nhau, ở bước này, ta xét các kết quả theo tiêu chuẩn phát biểu ban đầu là số LOOP ít nhất, nếu số LOOP bằng nhau, thì xét kết quả nào chứa LOOP có ít biến hơn thì tối giản hơn. Và có khả năng là chỉ có một công thức tối giản duy nhất hay nhiều công thức tương đương.
- Xét một ví dụ cụ thể cho giải thuật nêu trên:

  - Cho x = f(D, C, B, A) = Σ m (0, 1, 2, 3, 4, 5, 6, 8, 10, 11, 13, 15) với D là MSB, A là LSB

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Hình 69. Biểu đồ Karnaugh cho hàm 4 biến trên

- Bước 1: Liệt kê tất cả các LOOP của biểu đồ
- LOOP16: không có.
- LOOP8: không có.
- LOOP4: có 5 LOOP đặt tên là L4-1, L4-2, L4-3, L4-4, L4-5
- LOOP2: có 3 LOOP đặt tên là L2-1, L2-2, L2-3
- LOOP1: không có.
- Ta liệt kê cụ thể từng LOOP như sau

L4-1 là tập hợp 4 ô x_0, x_1, x_3, x_2 ⇨ L4-1 = {x_0, x_1, x_3, x_2} = $\overline{D}.\overline{C}$

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 51. L4-1 = $\overline{D}.\overline{C}$ chứa 4 ô trên một hàng

L4-2 là tập hợp 4 ô x_0, x_1, x_4, x_5 ⇨ L4-2 = {x_0, x_1, x_4, x_5} = $\overline{D}.\overline{B}$

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 52. L4-2 = $\overline{D}.\overline{B}$ chứa 4 ô tạo thành hình vuông

L4-3 là tập hợp 4 ô x_0, x_2, x_4, x_6 ⇨ L4-3 = {x_0, x_2, x_4, x_6} = $\overline{D}.\overline{A}$

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 53. L4-3 = $\overline{D}.\overline{A}$ chứa 4 ô ở hai cạnh

L4-4 là tập hợp 4 ô x_3, x_2, x_{11}, x_{10} ⇨ L4-4 = {x_3, x_2, x_{11}, x_{10}} = $\overline{C}.B$

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 54. L4-4 = $\overline{C}.B$ chứa 4 ô ở hai cạnh

L4-5 là tập hợp 4 ô x_0, x_2, x_8, x_{10} ⇨ L4-5 = {x_0, x_2, x_8, x_{10}} = $\overline{C}.\overline{A}$

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 55. L4-5 = $\overline{C}.\overline{A}$ chứa 4 ô ở 4 góc

L2-1 là tập hợp 2 ô x_5, x_{13} ⇨ L2-1 = {x_5, x_{13}} = $\overline{C}.\overline{B}.A$

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 56. L2-1 = $\overline{C}.\overline{B}.A$ chứa 2 ô kề nhau

L2-2 là tập hợp 2 ô x_{13}, x_{15} ⇨ L2-2 = {x_{13}, x_{15}} = D.C.A

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 57. L2-2 = D.C.A chứa 2 ô kề nhau

L2-3 là tập hợp 2 ô x_{15}, x_{11} ⇨ L2-3 = {x_{15}, x_{11}} = D.B.A

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 58. L2-3 = D.B.A chứa 2 ô kề nhau

- Bước 2: ô x_6 thuộc duy nhất L4-3 = $\{x_0, x_2, x_4, x_6\}$ = $\overline{D}.\overline{A}$ và ô x_8 thuộc duy nhất L4-5 = $\{x_0, x_2, x_8, x_{10}\}$ = $\overline{C}.\overline{A}$ ⇨ x = $\overline{D}.\overline{A}$ + $\overline{C}.\overline{A}$ + ... ở bước 2 ta đã tô được 6 ô, còn lại 6 ô x_1, x_3, x_5, x_{13}, x_{15}, x_{11} chưa tô

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 59. Kết quả tô biểu đồ Karnaugh sau bước 2

- Bước 3:

Bước 3.1:

Để tô những ô còn lại, ta chọn tuỳ ý một LOOP, trong trường hợp này ta chọn L4-1 = { x_0, x_1, x_2, x_3 } = $\overline{D}.\overline{C}$ ⇨ tô thêm 2 ô x_1, x_3 ⇨ x = $\overline{D}.\overline{A}$ + $\overline{C}.\overline{A}$ + $\overline{D}.\overline{C}$ + ... còn 4 ô x_5 , x_{13} , x_{11} , x_{15} chưa tô.

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 60. Kết quả tô biểu đồ Karnaugh sau bước 3.1

Bước 3.1.1:

Để tô những ô còn lại, ta chọn tuỳ ý một LOOP, trong trường hợp này ta chọn L2-3 = {x_{11} , x_{15}} = C.B.A ⇨ tô thêm 2 ô x_{11} , x_{15} ⇨ x = $\overline{D}.\overline{A} + \overline{C}.\overline{A} + \overline{D}.\overline{C}$ + C.B.A +… còn lại 2 ô x_5 , x_{13} chưa tô

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 61. Kết quả tô biểu đồ Karnaugh sau bước 3.1.1

Bước 3.1.1.1:

Để tô hai ô x_5 và x_{13} còn lại, ta chọn tuỳ ý L2-1 = {x_5, x_{13}} = D.$\overline{B}$.A ⇨ x = $\overline{D}.\overline{A} + \overline{C}.\overline{A} + \overline{D}.\overline{C}$ + C.B.A + D.$\overline{B}$.A (***<u>Công thức 1</u>***) và biểu đồ Karnaugh đã được tô xong hay ta đã có một công thức rút gọn.

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 62. Kết quả tô biểu đồ Karnaugh sau bước 3.1.1.1

Bước 3.1.1.2: (phản đề của 3.1.1.1)

Trong bước 3.1.1.1 việc lựa chọn L2-1 để tô hai ô x_5 và x_{13} là tùy ý, nên để tô 2 ô x_5 ,x_{13} còn lại mà không dùng L2-1, bắt buộc phải dùng L4-2 = {x_0 , x_1 , x_4, x_5} = $\overline{D}.\overline{B}$ và L2-2 = {x_{13} ,x_{15} } = D.C.A ⇨ x = $\overline{D}.\overline{A} + \overline{C}.\overline{A} + \overline{D}.\overline{C}$ + C.B.A + $\overline{D}.\overline{B}$ + D.C.A (***<u>Công thức 2</u>***)

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 63. Kết quả tô biểu đồ Karnaugh sau bước 3.1.1.2

Bước 3.1.2: (phản đề của 3.1.1)

Để tô thêm hai ô x_{11}, x_{15} mà không dùng L2-3 thì bắt buộc phải dùng L2-2 = $\{x_{13}, x_{15}\}$ = D.C.A và L4-4 = $\{x_2, x_3, x_{10}, x_{11}\}$ = $\overline{C}.B$ ⇨ $x = \overline{D}.\overline{A} + \overline{C}.\overline{A} + \overline{D}.\overline{C} + D.C.A + \overline{C}.B + \ldots$ còn lại 1 ô x_5 chưa tô

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 64. Kết quả tô biểu đồ Karnaugh sau bước 3.1.2

Bước 3.1.2.1:

Còn 1 ô x_5 chưa tô, ta chọn L4-2 = $\{x_0,x_1,x_4,x_5\}$ = $\overline{D}.\overline{B}$ để tô xong ⇨ $x = \overline{D}.\overline{A} + \overline{C}.\overline{A} + \overline{D}.\overline{C} + D.C.A + \overline{C}.B + \overline{D}.\overline{B}$ (***<u>Công thức 3</u>***)

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 65. Kết quả tô biểu đồ Karnaugh sau bước 3.1.2.1

Bước 3.1.2.2: (phản đề của 3.1.2.1)

Để tô ô x_5 mà không chọn L4-2 thì bắt buộc phải dùng L2-1 = $\{x_5, x_{13}\} = C.\overline{B}.A$ để tô xong ⇨ $x = \overline{D}.\overline{A} + \overline{C}.\overline{A} + \overline{D}.\overline{C} + D.C.A + \overline{C}.B + C.\overline{B}.A$ (***<u>Công thức 4</u>***)

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 66. Kết quả tô biểu đồ Karnaugh sau bước 3.1.2.2

Bước 3.2: (phản đề của 3.1)

Để tô được hai ô x_1, x_3 mà không chọn L4-1 thì bắt buộc phải chọn L4-2 = $\{x_0, x_1, x_4, x_5\} = \overline{D}.\overline{B}$ và L4-4 = $\{x_2, x_3, x_{10}, x_{11}\} = \overline{C}.B$ ⇨ $x = \overline{D}.\overline{A} + \overline{C}.\overline{A} + \overline{D}.\overline{B} + \overline{C}.B + \dots$ còn lại hai ô x_{13}, x_{15} chưa tô.

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 67. Kết quả tô biểu đồ Karnaugh sau bước 3.2

Bước 3.2.1:

Để tô hai ô x_{13}, x_{15} còn lại, ta chọn L2-2 = $\{x_{13}, x_{15}\} = D.C.A$ để tô xong ⇨ $x = \overline{D}.\overline{A} + \overline{C}.\overline{A} + \overline{D}.\overline{B} + \overline{C}.B + D.C.A$ (***<u>Công thức 5</u>***)

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 68. Kết quả tô biểu đồ Karnaugh sau bước 3.2.1

Bước 3.2.2 (phản đề của 3.2.1)

Để tô hai ô x_{13}, x_{15} còn lại, mà không chọn L2-2 = $\{x_{13},x_{15}\}$ = D.C.A thì phải chọn L2-1 = $\{x_5, x_{13}\}$ = $C.\bar{B}.A$ và L2-3 = $\{x_{11},x_{15}\}$ = D.B.A để tô xong ⇨ $x = \bar{D}.\bar{A} + \bar{C}.\bar{A} + \bar{D}.\bar{B} + \bar{C}.B + C.\bar{B}.A + D.B.A$ (***<u>Công thức 6</u>***)

DC \ BA	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5		x_6
11		x_{13}	x_{15}	
10	x_8		x_{11}	x_{10}

Bảng 69. Kết quả tô biểu đồ Karnaugh sau bước 3.2.2

Bước 4:

Sau bước 3 ta có sáu công thức:

$x = \bar{D}.\bar{A} + \bar{C}.\bar{A} + \bar{D}.\bar{C} + C.B.A + D.\bar{B}.A$ (***<u>Công thức 1</u>***)

$x = \bar{D}.\bar{A} + \bar{C}.\bar{A} + \bar{D}.\bar{C} + C.B.A + \bar{D}.\bar{B} + D.C.A$ (***<u>Công thức 2</u>***)

$x = \bar{D}.\bar{A} + \bar{C}.\bar{A} + \bar{D}.\bar{C} + D.C.A + \bar{C}.B + \bar{D}.\bar{B}$ (***<u>Công thức 3</u>***)

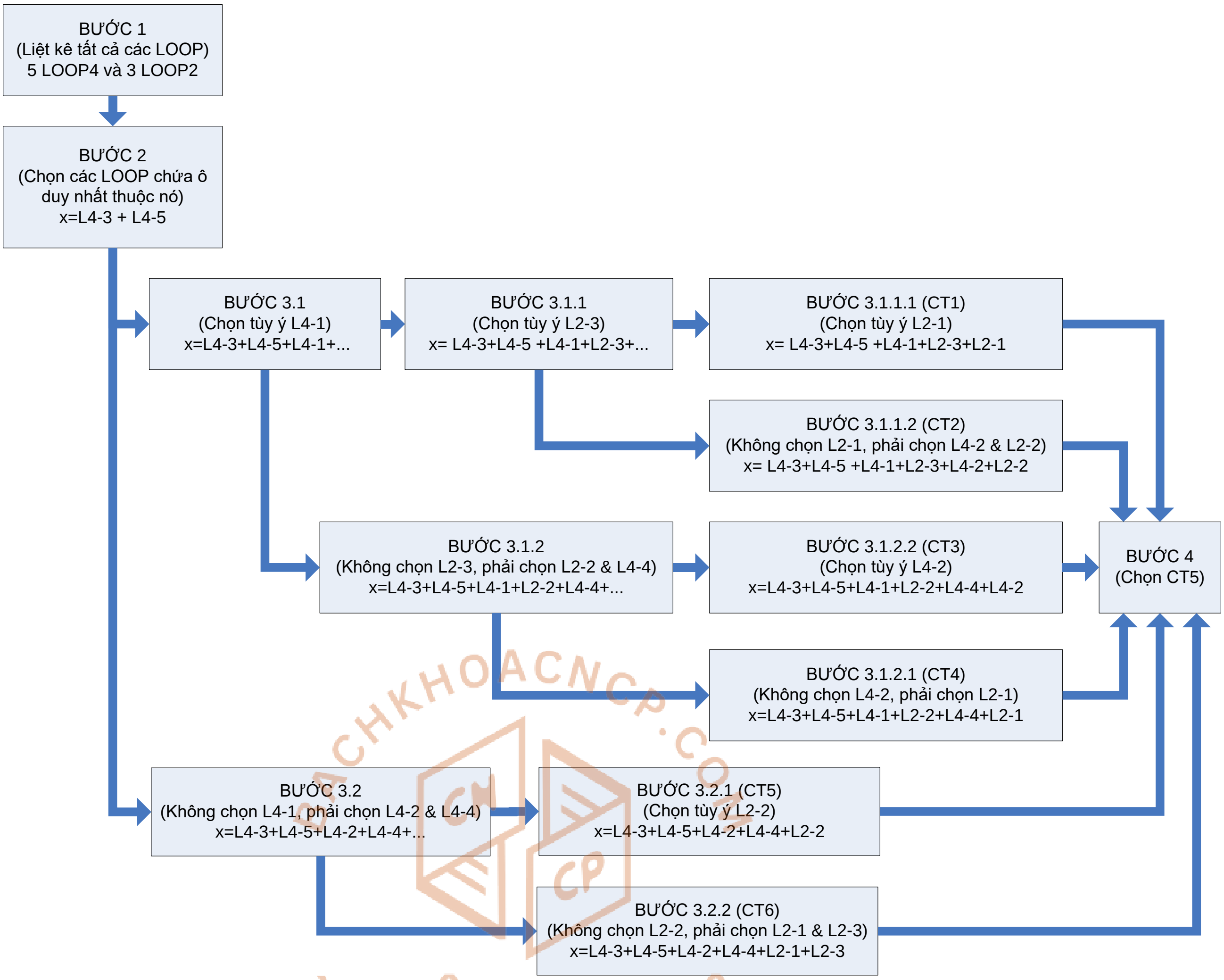
$x = \bar{D}.\bar{A} + \bar{C}.\bar{A} + \bar{D}.\bar{C} + D.C.A + \bar{C}.B + C.\bar{B}.A$ (***<u>Công thức 4</u>***)

$x = \bar{D}.\bar{A} + \bar{C}.\bar{A} + \bar{D}.\bar{B} + \bar{C}.B + D.C.A$ (***<u>Công thức 5</u>***)

$x = \bar{D}.\bar{A} + \bar{C}.\bar{A} + \bar{D}.\bar{B} + \bar{C}.B + C.\bar{B}.A + D.B.A$ (***<u>Công thức 6</u>***)

Ta chọn công thức 5 vì đó công thức rút gọn nhất.

- Giải thuật dùng biểu đồ Karnaugh cho hàm 4 biến nói trên được minh họa bằng lưu đồ sau:

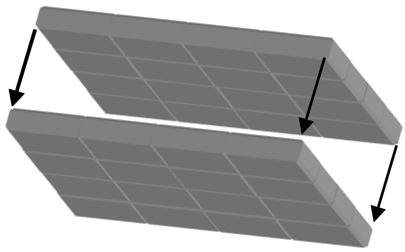


Hình 70. Lưu đồ cho giải thuật rút gọn hàm 4 biến đã cho bằng biểu đồ Karnaugh.

1.12.5- Biểu đồ Karnaugh cho hàm 5 biến x = f(E, D, C, B, A) và hàm 6 biến x = f(F, E, D, C, B, A)

1.12.5.1- Biểu đồ Karnaugh cho hàm 5 biến

- Ta có thể mở rộng phạm vi áp dụng biểu đồ Karnaugh lên đến hàm 5 biến và hàm 6 biến. Biểu đồ Karnaugh cho hàm 5 biến là hai biểu đồ Karnaugh cho hàm 4 biến chồng lên nhau, gồm 32 ô. Như vậy, mỗi ô sẽ kề với 5 ô như hình vẽ dưới:



Bảng 70. Biểu đồ Karnaugh cho hàm 5 biến

Trong mặt phẳng, ta lật biểu đồ Karnaugh của hàm 5 biến thành hai biểu đồ Karnaugh cho hàm 4 biến nằm cạnh nhau như hình vẽ:

	E				$\overline{E}$				
	$B.\overline{A}$	$B.A$	$\overline{B}.A$	$\overline{B}.\overline{A}$	$\overline{B}.\overline{A}$	$\overline{B}.A$	$B.A$	$B.\overline{A}$	
$\overline{D}.\overline{C}$	x_{18}	x_{19}	x_{17}	x_{16}	x_0	x_1	x_3	x_2	$\overline{D}.\overline{C}$
$\overline{D}.C$	x_{22}	x_{23}	x_{21}	x_{20}	x_4	x_5	x_7	x_6	$\overline{D}.C$
$D.C$	x_{30}	x_{31}	x_{29}	x_{28}	x_{12}	x_{13}	x_{15}	x_{14}	$D.C$
$D.\overline{C}$	x_{26}	x_{27}	x_{25}	x_{24}	x_8	x_9	x_{11}	x_{10}	$D.\overline{C}$

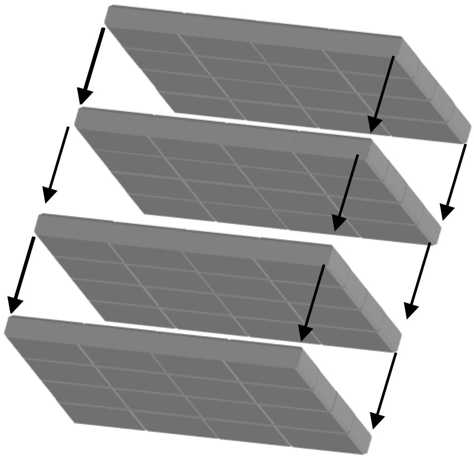
Bảng 71. Biểu đồ Karnaugh cho hàm 5 biến trên mặt phẳng

- Khi chiếu lên mặt phẳng, thì chúng ta có hai biểu đồ Karnaugh đối xứng gương qua trục giữa, những ô kề nhau thuộc bên E hay $\overline{E}$ thì cách tính không có gì thay đổi. Riêng những ô kề nhau thuộc bảng bên kia thì lưu ý là đối xứng gương như ô x_0 sẽ kề ô x_{16}, ô x_5 kề với ô x_{21}, x_{15} kề với ô x_{31}, x_{10} kề với x_{26}

- Vậy 8 ô có giá trị $x_i = 1$ trên cùng một hàng thì rút gọn được 3 biến, 16 ô có giá trị $x_i = 1$ trên hai hàng kề nhau thì rút gọn được 4 biến, 8 ô thuộc hai cột đối xứng gương qua trục giữa có giá trị $x_i = 1$ sẽ rút gọn được 3 biến, 16 ô thuộc 4 cột kề nhau sẽ rút gọn được 4 biến

1.12.5.2- Biểu đồ Karnaugh cho hàm 6 biến là:

- Biểu đồ Karnaugh cho hàm 6 biến là một hình lập phương gồm 4 biểu đồ Karnaugh cho hàm 4 biến chồng lên nhau, gồm 64 ô. Mỗi ô sẽ kề với 6 ô như hình vẽ:



Bảng 72. Biểu đồ Karnaugh cho hàm 6 biến

Trong mặt phẳng, ta lật biểu đồ Karnaugh của hàm 6 biến thành bốn biểu đồ Karnaugh cho hàm 4 biến nằm cạnh nhau như hình vẽ:

	$\overline{F}.E$				$\overline{F}.\overline{E}$				
	$B.\overline{A}$	$B.A$	$\overline{B}.A$	$\overline{B}.\overline{A}$	$\overline{B}.\overline{A}$	$\overline{B}.A$	$B.A$	$B.\overline{A}$	
$\overline{D}.\overline{C}$	x_{18}	x_{19}	x_{17}	x_{16}	x_{0}	x_{1}	x_{3}	x_{2}	$\overline{D}.\overline{C}$
$\overline{D}.C$	x_{22}	x_{23}	x_{21}	x_{20}	x_{4}	x_{5}	x_{7}	x_{6}	$\overline{D}.C$
$D.C$	x_{30}	x_{31}	x_{29}	x_{28}	x_{12}	x_{13}	x_{15}	x_{14}	$D.C$
$D.\overline{C}$	x_{26}	x_{27}	x_{25}	x_{24}	x_{8}	x_{9}	x_{11}	x_{10}	$D.\overline{C}$
$D.\overline{C}$	x_{58}	x_{59}	x_{57}	x_{56}	x_{40}	x_{41}	x_{43}	x_{42}	$D.\overline{C}$
$D.C$	x_{62}	x_{63}	x_{61}	x_{60}	x_{44}	x_{45}	x_{47}	x_{46}	$D.C$
$\overline{D}.C$	x_{54}	x_{55}	x_{53}	x_{52}	x_{36}	x_{37}	x_{39}	x_{38}	$\overline{D}.C$
$\overline{D}.\overline{C}$	x_{50}	x_{51}	x_{49}	x_{48}	x_{32}	x_{33}	x_{35}	x_{34}	$\overline{D}.\overline{C}$
	$B.\overline{A}$	$B.A$	$\overline{B}.A$	$\overline{B}.\overline{A}$	$\overline{B}.\overline{A}$	$\overline{B}.A$	$B.A$	$B.\overline{A}$	
	$F.E$				$F.\overline{E}$				

Bảng 73. Biểu đồ Karnaugh cho hàm 6 biến trên mặt phẳng

- Cách tính các ô kề nhau cũng giống như biểu đồ Karnaugh hàm 5 biến, lưu ý đến việc đối xứng gương qua các trục và đối xứng qua tâm toạ độ

- Giải thuật rút gọn cho biểu đồ Karnaugh của hàm 5 và 6 biến cũng tương tự như hàm 4 biến nêu trên.

1.13- Rút Gọn Một Biểu Thức Bằng Phương Pháp Biểu Đồ Karnaugh – Dạng Chuẩn Hội

- Về nguyên tắc, thì rút gọn một biểu thức logic bằng phương pháp biểu đồ Karnaugh dạng chuẩn hội cũng giống như dạng chuẩn tuyển, điều khác biệt duy nhất là tính những giá trị $x_i = 0$ thay vì chọn $x_i =1$
- Để tổng quát cách xây dựng biểu đồ Karnaugh, ta đưa ra một số nguyên tắc sau cho biểu đồ Karnaugh 4 biến dạng chuẩn hội:

  - Biểu đồ Karnaugh cho hàm 4 biến là 1 hình vuông có 16 ô.
  - Mỗi ô ứng với 1 dòng trong bảng thực trị (tích của 4 biến D, C, B, A và x_i).
  - Mỗi ô nằm kế cận 4 ô, như ô x_5 kề 4 ô x_1, x_4, x_7, x_{13}, ô x_6 kề với 4 ô x_2, x_4, x_7, x_{14}, ô x_0 kề 4 ô x_1, x_4, x_8, x_{10}, ...
  - Hai ô cạnh nhau chỉ khác nhau duy nhất 1 bit.
  - LOOP là tập hợp các ô có giá trị $x_i = 0$ nằm cạnh nhau theo nguyên tắc 2^n, với $n = 0, 1, 2, 3, 4$ ⇨ LOOP1, LOOP2, LOOP4, LOOP8, LOOP16.
  - Ô nằm riêng lẻ một mình có giá trị $x_i = 0$, khoanh chúng lại thành một ***vòng*** gồm 1 ô, được gọi là LOOP1, không rút gọn được biến nào.
  - Hai ô nằm cạnh nhau có giá trị là $x_i = 0$, khoanh chúng lại thành một ***vòng*** gồm 2 ô, được gọi là LOOP2, thì rút gọn được một biến.
  - Bốn ô nằm cạnh nhau có giá trị là $x_i = 0$, khoanh chúng lại thành một ***vòng*** gồm 4 ô, được gọi là LOOP4, thì rút gọn được hai biến. Trong biểu đồ Karnaugh 4 biến, LOOP4 là 4 ô tạo thành hình vuông, hay tạo thành một cột, một hàng hay 4 ô ở 4 góc.
  - Tám ô nằm cạnh nhau có giá trị là $x_i = 0$, khoanh chúng lại thành một ***vòng*** gồm 8 ô, được gọi là LOOP8, thì rút gọn được ba biến. Trong biểu đồ Karnaugh 4 biến, LOOP8 là 8 ô tạo thành hai hàng hay hay cột.

- Mười sáu ô nằm cạnh nhau có giá trị là $x_i = 0$, khoanh chúng lại thành một ***vòng*** gồm 16 ô, được gọi là LOOP16, thì rút gọn được bốn biến.

- Cách biểu diễn biểu đồ Karnaugh cho hàm 4 biến dùng trong dạng chuẩn hội

BA / DC	00	01	11	10
00	x_0	x_1	x_3	x_2
01	x_4	x_5	x_7	x_6
11	x_{12}	x_{13}	x_{15}	x_{14}
10	x_8	x_9	x_{11}	x_{10}

Bảng 74. Cách biểu diễn biểu đồ Karnaugh dùng trong tài liệu

- Và để cho gọn, những ô có giá trị $x_i = 1$ sẽ để trống, chỉ đưa vào những ô có giá trị $x_i = 0$.

- Hàng số 1 là $D + C$, hàng số 2 là $D + \overline{C}$, hàng số 3 $\overline{D} + \overline{C}$, hàng số 4 là $\overline{D} + C$.

- Cột số 1 là $B + A$, cột số 2 là $B + \overline{A}$, cột số 3 là $\overline{B} + \overline{A}$, cột số 4 là $\overline{B} + A$

- Phương pháp rút gọn giống như trong trường hợp dạng chuẩn tuyển

1.14- Rút Gọn Một Biểu Thức Bằng Phương Pháp Biểu Đồ Karnaugh – Có Những Giá Trị Của Hàm Là Tùy Chọn

- Trong trường hợp biểu đồ Karnaugh có 1 giá trị x_i là tùy chọn, có nghĩa là chọn $x_i = 0$ hay chọn $x_i = 1$ đều được, thì ta phải phân tích thành hai biểu đồ Karnaugh, một biểu đồ Karnaugh ứng với $x_i = 0$ và một biểu đồ Karnaugh ứng với $x_i = 1$. Nếu có hai giá trị x_i, x_j là tuỳ chọn thì ta lập thành 4 biểu đồ Karnaugh ứng với các cặp $x_i, x_j = 0{,}0$ hay $x_i, x_j = 0{,}1$ hay $x_i, x_j = 1{,}0$ hay $x_i, x_j = 1{,}1$. Nếu có N giá trị x_i là tuỳ chọn thì ta sẽ có 2^N biểu đồ Karnaugh để rút gọn.

- Sau khi tách ra các trường hợp, thì ta rút gọn từng trường hợp một để tìm ra công thức rút gọn nhất.